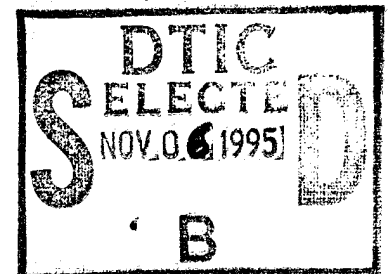FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

DATA ASSISTED KNOWLEDGE ACQUISITION

FOR ALTERATION OF

TABLE-BASED EXPERT NETWORKS

By

ROBERT G. TIMPANY

A Thesis submitted to the
Department of Computer Science
in partial fulfillment of the
requirements for the degree of
Master of Science
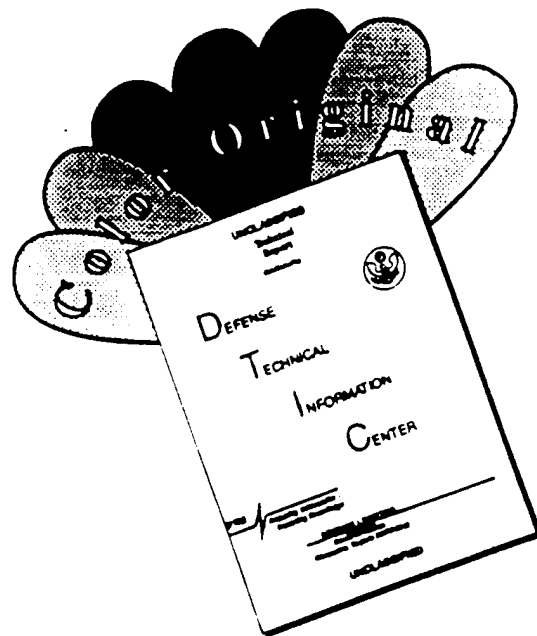
Degree Awarded:
Summer Semester, 1995

19951103 025

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.
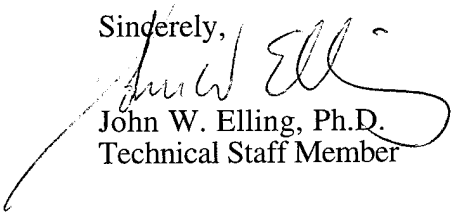
Los Alamos
N A T I O N A L   L A B O R A T O R Y

Engineering Sciences & Applications Division
ESA-MT, Measurement Technology   MS J580
Test, Measurement, & Automation Solutions
Los Alamos, New Mexico 87545
(505) 667-4316
FAX (505) 665-3911

Date: October 30, 1995
Refer to: ESA-MT-95-391

Defense Technical Information Center
Attn: Chester Brooks
8725 John J. Kingman Road
Suite 0944
Fort Belvoir, VA 22060-6218

The document "Data Assisted Knowledge Acquisition for Table-Based Expert

Networks" by Robert G. Timpany, is approved for public release, distribution

unlimited. The document was generated at Florida State University for the

Department of Energy in the performance of contract number C87-101395 002.

Sincerely,

John W. Elling, Ph.D.
Technical Staff Member

Cys:
CIC-10, MS A150
ESA-MT File

FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

DATA ASSISTED KNOWLEDGE ACQUISITION

FOR ALTERATION OF

TABLE-BASED EXPERT NETWORKS

By
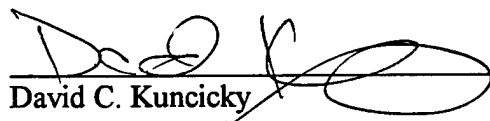
ROBERT G. TIMPANY

A Thesis submitted to the
Department of Computer Science
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Summer Semester, 1995
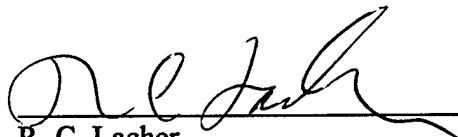
The members of the Committee approve the thesis of Robert G. Timpany defended on June 27, 1995.

Susan I. Hruska
Professor Directing Thesis

David C. Kuncicky
Committee Member

R. C. Lacher
Committee Member

Approved:

R. C. Lacher, Chair, Department of Computer Science

Dedicated to the symbolism contained in every red stripe of our flag: the men and women who, as fellow citizens, made the supreme sacrifice to purchase the freedom we take for granted today.

Acknowledgements

I would like to sincerely thank my Major Professor for taking me on as a student, enlisting me into the CAEN team and providing the guidance needed to succeed. I would also like to thank my committee members for their efforts in aiding my work.

I would like to personally thank those teachers who had a significant impact on my efforts at Florida State: Dr. Stephen Leach, Dr. David Whalley, Dr. Charles Kacmar, Dr. Gregory Riccardi, and Dr. Theodore Baker. I would like to especially thank Dr. David Kuncicky for the two exceptional classes he instructed.

Thanks to my fellow CAEN project members and true friends Kristin Adair and Alan Levis. My thanks to Tom Berrisford who provided significant input concerning network alteration techniques.

I would also like to acknowledge the friendships built here at Florida State that helped steady my course during some dark hours: Buck Surdu, Jerry Franke, Joe Pasko, Wayne

Sprague (I'll still tap his phone), Jason Orendorf, and all the other graduate students in the department. I would like to personally thank Cathie LeBlanc for helping me recognize some of the sharper edges to my life.

Thanks to my family, and in particular my father, for instilling in me those values needed to succeed in life. My deepest thanks to the driving influence and utmost joy in my life, my son Ryan.

I would like to thank Jim, Cliff, and others like them who made, lived, and know the true meaning of *Special*. Thanks also to Alan, Perry, Joe, Jeff, Bob G., Leon, Randy, John, Rick, Bob W., and all others who follow *De Oppresso Liber* today. Most of all I would like to thank our Country for allowing me to serve her.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Knowledge-based systems simulate the decision making process of human experts. Two main approaches to knowledge-based systems are *expert systems* and *network systems*. Researchers at Florida State University developed *expert networks* to combine the benefits of expert systems and network systems, including the explanation capability of expert systems and the learning capability of network systems. Expert networks still rely on the knowledge engineer's ability to determine the connection relationships among inputs, the set of stored facts and rules, and conclusions.

The interpretation of gas chromatography data provides the application domain of this research. The *Contaminant Analysis Expert Network* project involves applying expert network technology to this problem domain in which analytical chemists use knowledge tables to represent the domain knowledge. The work of this thesis centers on the need to build automated systems that utilize example data to aid in knowledge acquisition and structural alteration of these table-based expert networks.

The new approach to network structural alteration presented here automates the ability to confirm, refine, and augment expert knowledge. The automated tool developed in this thesis combines the knowledge derived from experts with the relationships discovered in data. In altering the expert networks to improve prediction performance, the tool maintains the ability to retrieve the expert knowledge from the network. The result is expert networks

which structurally capture both the table-based knowledge of analytical chemists and information gleaned from example data. These networks outperform networks built from expert knowledge alone.

CHAPTER 1

INTRODUCTION

Knowledge-based systems simulate the decision making process of human experts. Two main approaches to knowledge-based systems are *expert systems* and *network systems*. Expert systems usually involve an inference mechanism, a set of stored facts and rules, and a user interface. The expert system's inference mechanism takes information from a user, searches the set of stored facts and rules for those that apply, and arrives at a conclusion. Expert systems typically provide an explanation facility to illustrate how the inference mechanism arrives at the conclusion.

Expert systems are only as accurate as the expert information in the set of stored facts and rules. In problem domains with conflicting or incomplete expert knowledge, the performance of expert systems suffers.

Artificial neural network systems utilize many interconnected processing units organized in layers to reach a conclusion. These processing units relate to one another through a connection architecture, typically depicted by directed graphs. Nodes, or vertices, in the graphs are the processing units and the edges are the connections. The processing units receive some form of empirical information from either the user interface or a preceding layer. Based on the functionality of the processing units, the units transfer a value onto their outgoing edges. Network systems reach a conclusion based on the evaluation of output

values in the output (or final) layer of the network.

Each connection in the network architecture has a corresponding weight. A learning algorithm adjusts these weights to improve the accuracy of the network. Network systems provide a means of capturing expert knowledge through the learning process. However, the captured expert knowledge is difficult to extract in the form of rules that the user could understand.

Researchers at Florida State University developed *expert networks* to combine the benefits of expert systems and network systems [Lacher, 1993]. Expert network technology provides a mapping from a rule-based expert system to a network architecture [Kuncicky, Hruska and Lacher, 1992]. Training alters the network to improve prediction accuracy. The expert network technology provides a second mapping, the inverse of the first, from the network back to an expert system. Thus, expert network technology combines the explanation capability of expert systems with the learning capability of network systems. Expert networks still rely on the expert's ability to determine the connection relationships among inputs, the set of stored facts and rules, and conclusions. The ability to automatically discover new rules or connections in expert networks is an outstanding need.

The interpretation of *Gas Chromatography* (GC) data provides the domain of the problems reported in this thesis. The *Contaminant Analysis Expert Network* (CAEN) project encompasses applying expert network technology to this problem domain[1]. Although there

---

[1] CAEN is a Gensym G2-based [Gensym, 1993] Expert Network project to aid in the Department of Energy's Contaminant Analysis Automation project. Team members include Susan Hruska, Alan Levis, Robert Timpany, and Kristin Adair. CAEN team affiliates include R.C. Lacher and Douglas Klotter. The use of the acronym CAEN indicates either the team or the network based on context.

are many analytical chemists with expertise in this domain, there is little formal specification of their knowledge. The lack of formal specification of expert knowledge results in an expert network with poor prediction capability. Poor prediction capability calls for an effort to improve the structure of expert networks used in the CAEN project in a manner that allows confirmation, refinement, and augmentation of expert knowledge.

Several methods exist to automate network creation and alteration in order to improve prediction results. These network alteration methods generally do not have the ability to explain why the method adds, deletes, or changes a particular connection. The lack of an explanation capability runs counter to the CAEN project's goals, and so precludes the use of existing network alteration methods.

The research reported in this thesis provides a new approach to the problem of network alteration to improve prediction results. The automated tool created as part of this research, called *NetMedic*, can confirm, refine, and augment expert knowledge. NetMedic combines the knowledge derived from experts with the relationships discovered in data. This process facilitates the alteration of an expert network to improve performance while still maintaining the ability to retrieve the expert knowledge from the network. Use of NetMedic resulted in CAEN networks with better prediction capability than those CAEN networks developed through dialogue with experts alone.

Importance of the Problem

The United States Department of Energy (DOE) is responsible for cleaning up many hazardous waste sites resulting from weapons research and nuclear power research. The majority of the cleanup efforts involve testing and disposal of buried waste. The Contaminant

Analysis Automation (CAA) project grew from the need to reliably process high volumes of soil analyses at each site in a restricted time period [Elling, Klatt and Unruh, 1994]. This project includes efforts of scientists and engineers from Los Alamos National Laboratory (LANL), Idaho National Engineering Laboratory (INEL), Oak Ridge National Laboratory (ORNL), Sandia National Laboratory (SNL), Varian Equipment Corporation, University of New Mexico, University of Florida, and Florida State University, among others. The overall goal of CAA is to field a fully automated system that includes robot samplers, automated preparation, data interpretation, and analysis.

The CAEN team works with the two main components of CAA: the Analytical Instrument Module (AIM) and the Data Interpretation Module (DIM). The AIM provides an automated process for sample preparation and data collection. The DIM then evaluates the raw data, performs analyses, validates results, and provides quality control. There is also a higher level module that controls the entire process, the Task Sequence Controller (TSC) [Elling, Klatt and Unruh, 1994].

Currently, contaminant analysis involves humans processing soil samples in a largely manual fashion. The samples require a good deal of preparation and many complicated tests. Gas chromatography analysis provides a vehicle for discerning contaminants in soil samples [Zlatkis and Poole, 1981]. An automated gas chromatography system prepares the sample and then heats it until gaseous. The equipment mixes the gaseous sample with a carrier gas. The flow of the carrier gas in the equipment carries the gaseous sample mixture past a detector. The light absorption profile yields base element analysis. Laboratories typically conduct the soil tests as overnight batch runs.

To analyze gas chromatograms at the completion of the run, an expert must examine each chromatogram and correlate peak shape, placement, baseline, and amplitude to a given element. The combination of peaks in a chromatogram yield a set of elements that provides a signature for a given compound. Interpretation of chromatograms is difficult, and can be impossible with flawed chromatograms. Flawed chromatograms can be the result of machine faults or sample preparation problems. Since the jobs are run in batch, these failures tend to go unrecognized until the morning. These unrecognized failures force the reevaluation of many samples with considerable loss of money, time, and effort.

Automated Data Interpretation

The first responsibility of the DIM is to evaluate the raw GC data to determine whether or not the data is analyzable. The evaluation involves scanning the chromatogram for recognizable failure signatures. Analytical chemists grouped these failure signatures into a set of *symptoms*. These symptoms (Appendix A) provide clues to problems with the equipment or sample preparation. Experts also grouped the problems associated with the failure signatures into a set of *causes* (Appendix A). Depending on the amount and severity of symptoms present in the chromatogram, the DIM decides whether the chromatogram is analyzable by an expert. The decision on analyzability involves the determination of whether an expert could accurately determine the contaminant in the sample from the chromatogram.

For example, *No Peaks* (Appendix A) occurs when there is a *Leaking Syringe* (Appendix A). With a *Leaking Syringe*, the GC instrument produces a chromatogram with a flat line at the base. The lack of information in the resultant chromatogram makes further analysis of the sample impossible. The DIM should detect *No Peaks* symptom, conclude

Leaking Syringe, and notify the TSC. The TSC, in turn, halts the assembly line of samples and either fixes the problem automatically or notifies the operator of the problem and gives a recommended solution.

Lahiri and Stillman tackled the problem of automating determination of analyzable GC data using conventional expert systems [Lahiri and Stillman, 1992]. Lahiri and Stillman related the symptoms to causes through the use of a truth table, with promising early results.

The CAEN Approach

In 1993, Jamie Ferguson, a senior engineer at INEL, proposed using the expert network technology in cooperation with researchers at Florida State University to enhance the GC data interpretation. After a meeting between the DIM project leader, John Elling (LANL), Jamie Ferguson, and Susan Hruska that same year, the CAEN project at Florida State University began in earnest.

As a first project task, the CAEN team is building a system which decides whether a given gas chromatogram is analyzable. The CAEN project uses a *knowledge table* to represent the expert's knowledge in predicting machine and sample failures from symptoms extracted from gas chromatograms. These relationships differ greatly from the standard truth table representation employed by Lahiri and Stillman. Each cell in the knowledge table contains an entry corresponding to the type of relationship between the symptoms and causes. Blank cells in the knowledge table indicate no relationship between the symptoms and causes; linguistic qualifiers capture the expert's view of the same.

The knowledge captured informally represents an expert system using the table as the knowledge base and the human brain as the inferencing mechanism. Another major task of

the CAEN team is the determination of an *inference mechanism* for the proposed network based on this captured information.

During the knowledge acquisition process, the CAEN team identified wide discrepancies in the experts' knowledge. The CAEN team draws the information base for this problem domain from the personal experience of each chemist and trouble-shooting manuals provided by the GC instrument manufacturers. The information base varies greatly from chemist to chemist. In addition, differences in instrumentation manufacturer, configuration, and sample type provide different relationships in the table.

NetMedic Overview

The first network built by the CAEN team from the knowledge table performed poorly on real data. Although the inference mechanism for the network was still being refined, the structural relationships in the knowledge table needed the most attention. Lack of confidence in the architecture of the knowledge table prompted the need for an algorithm to *confirm*, *refine*, and *augment* the expert's knowledge. NetMedic is the response to this need.

NetMedic gathers statistics on the symptom and cause data. NetMedic confirms expert knowledge when the data analysis suggests the same connection as the expert. It refines expert knowledge when the statistics recommend a different type of connection than already exists. Augmentation occurs when NetMedic suggests either a deletion of a connection or the addition of a new one. These additions and deletions may be due to the discovery of new knowledge or indications of incorrect inputs.

NetMedic produces its statistics from a set of data files with symptom and cause information. The CAEN network must preprocess the data to provide the cause values and

results. The CAEN networks currently use a symptom file format described in Appendix B. Appendix C illustrates the format for NetMedic's input file. NetMedic records the frequency, mean, and standard deviation of symptom values as they relate to a particular cause, and proposes connections based on these statistics.

NetMedic normally runs in an interactive mode. It prompts the expert to make decisions when recommending a particular refinement or augmentation. The expert may accept or reject each of NetMedic's suggestions. NetMedic has a second mode used when the expert is not available. In this automatic mode, NetMedic creates connections directly from statistical relationships in the data.

Contributions and Assumptions

NetMedic's ability to discover new knowledge is important to the process of building knowledge tables which reflect both the expertise of analytical chemists and information from sample data. NetMedic has already discovered several new relationships between symptoms and causes. The signal processing team, Randy Roberts and Sharbari Lahiri of LANL, received valuable feedback on the accuracy of their symptom detection algorithms using NetMedic, as discussed in Chapter 7. NetMedic also provides better base networks for the expert network back propagation training developed by the CAEN team (Chapter 5).

Use of NetMedic carries the assumption that the experts have identified all symptoms and causes pertinent to the GC analysis problem. Furthermore, the data set must be representative of the entire spectrum of problems for a particular GC instrument and configuration.

NetMedic provides an interface between human experts and sample GC data to allow effective structural alteration of expert networks and knowledge acquisition. After confirming, refining, and augmenting, experts can still retrieve the knowledge from the network in comprehensible form. NetMedic produces CAEN networks that outperform networks developed by experts alone.

CHAPTER 2

DECISION MAKING SYSTEMS

A critical decision in the Contaminant Analysis Expert Network (CAEN) project was deciding upon an approximate reasoning method for representing and reasoning with the expert knowledge for Gas Chromatography (GC) analysis. This chapter briefly discusses two main paradigms, expert systems and neural networks, a hybrid extension called expert networks, and other major theories of representing knowledge and dealing with uncertainty. The CAEN team considered these decision making systems when designing the inference mechanism discussed in Chapter 3.

Expert Systems

Expert systems have as a base some amount of collected expert knowledge. This knowledge is often represented in the form of rules. Expert systems generally take some form of input from a user (human or machine), and provide advice in some manner to the requester. An inference engine combines the stored facts, rules, and inputs to reach a decision. Figure 1 shows the typical layout of an expert system. Experts traditionally find the expression of knowledge in rules easy to use. The explanation facilities which accompany many expert systems enhances the credibility of decisions made by these systems.

```
┌─────────────────────────────────────────────┐
│                  ┌──────────────┐            │
│                  │    User      │            │
│                  │  Interface   │            │
│                  └──────────────┘            │
│                       ↕                      │
│                  ┌──────────────┐            │
│                  │  Inference   │            │
│                  │   Engine     │            │
│                  └──────────────┘            │
│            ↙                    ↘            │
│   ┌──────────────┐      ┌──────────────┐     │
│   │ Stored facts │      │ Explanation  │     │
│   │  and rules   │      │   Facility   │     │
│   └──────────────┘      └──────────────┘     │
└─────────────────────────────────────────────┘
```
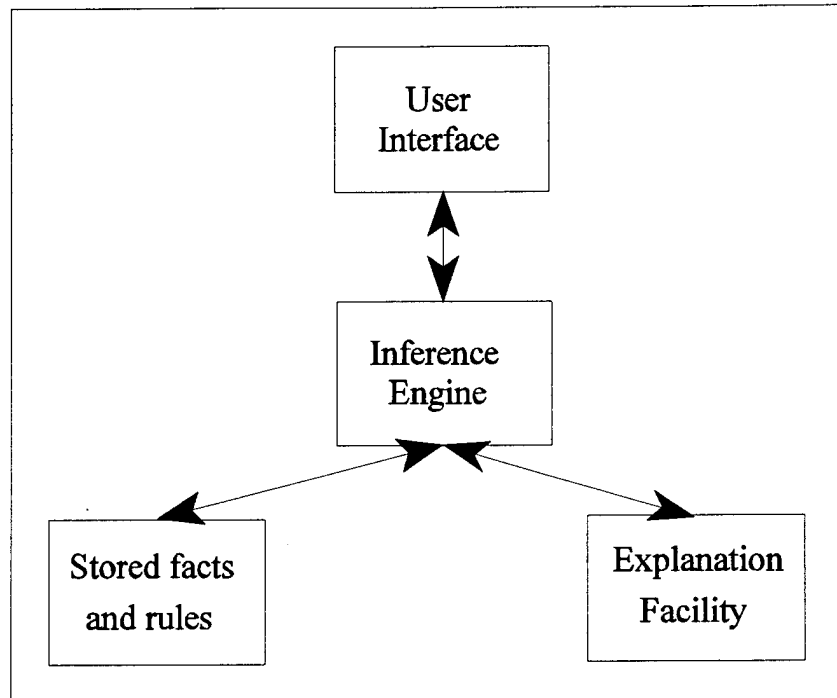
Figure 1: Typical Layout for an Expert System.

Expert systems are usually highly domain dependent. Years of training which go into making an expert can be captured and distributed to a wide audience by mass producing the expert systems in software. These systems provide this expertise at a great cost saving. Expert systems programmed with the advice of multiple experts combine knowledge sources to make better decisions.

Artificial Neural Network Systems

Artificial neural networks are another type of decision making system. An artificial neural network contains many interconnected processing elements. Theories of what we know of neurons found in our brains inspired the development of the processing elements. Each element receives input in the form of a value transmitted along a connection. Processing elements may be of different types and functionality. Connections between processing

elements have weights associated with them. Depending on the style of network, a learning process may alter some or all of these weights. This learning trains the network to associate a particular pattern of inputs with an output pattern.

Figure 2 depicts a typical neural network with a layered, feedforward topology. Networks may utilize either a supervised learning method or an unsupervised learning method. After an input pattern is presented and processed by the network, the supervised learning method compares the resulting output pattern (which may be a single value)
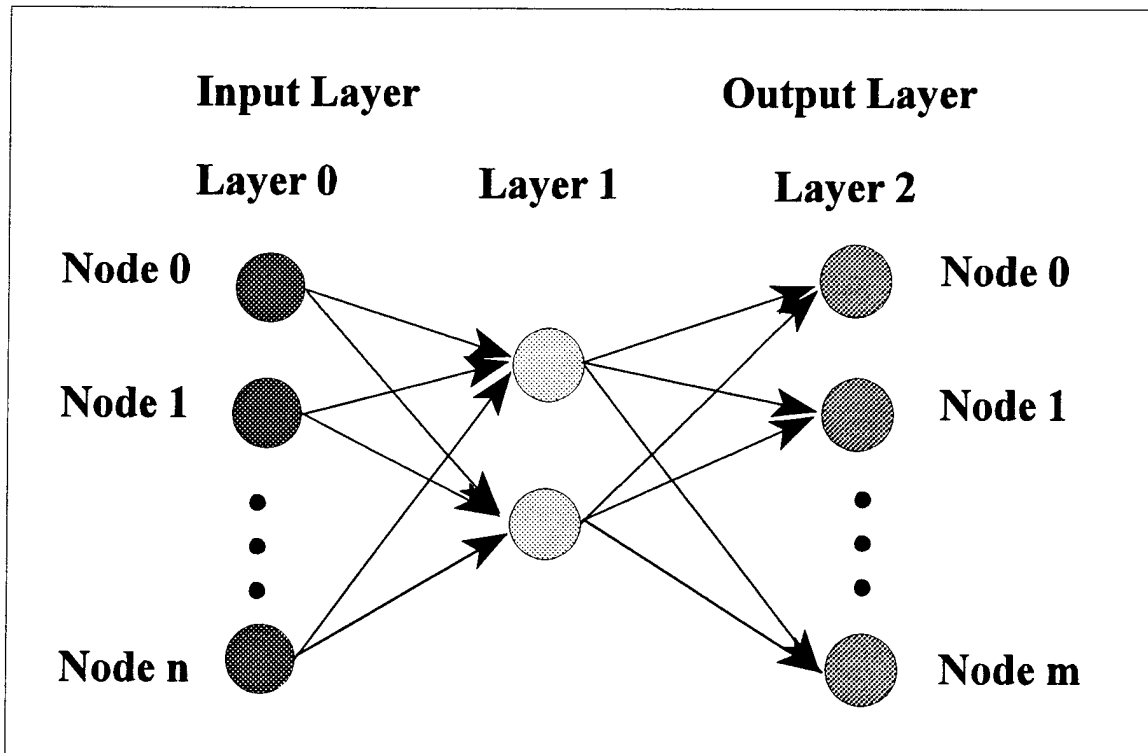
Figure 2: **A Layered Feedforward Neural Network.**

with a known output pattern that represents the correct answer for this input pattern. The learning method calculates the pattern error and updates the weights in the network in a manner to decrease the error in the network. This learning algorithm repeats this process until the output reaches an error tolerance. In an unsupervised method, the matching output

pattern is unknown. This style of network must be able to learn to cluster patterns together without any examples of "ideal" outputs.

The major drawback with this style approach is the lack of an explanation facility. It is difficult to attach meaning to why the network reaches a particular decision.

Expert Networks

Kuncicky, Hruska, and Lacher combined the explanatory capabilities of a rule-based system with the learning capability of artificial neural networks in *expert networks* [Kuncicky, Hruska, and Lacher, 1992]. Expert networks provide a mapping from a rule-based expert system to a type of artificial neural network and back. The mapping allows training with data while the system is in network form. The end result is an expert system tuned to increase its prediction accuracy. Expert networks retain the facility of the expert system to explain its logic chain.

In the translation of expert system to neural network, rules map to a graph by viewing the knowledge base as a collection of assertions which make up the consequents and antecedents of implications (rules). These assertions map to vertices (or nodes) in a directed graph. The graph edges connect the premises of rules to their conclusion. The certainty factors associated with the original rules map to weights along the output edges of the nodes. The inference engine of the expert system provides the firing functions for the nodes.

The implication of the mapping from an expert system to a network is that one can use the neural network style learning algorithm to modify the certainty of each rule in the original rule base. The rules themselves are unchanged. Therefore, the relationships between evidence and conclusions remain constant although the certainties of the conclusions do not.

Kuncicky and colleagues at Florida State University demonstrate the application of expert network technology through the Wine Advisor network [Kuncicky, Hruska, and Lacher, 1992]. In expert networks, neural network training refines the prediction capability of the original expert system. The ability to refine expert knowledge while retaining the critical relationship information was a key factor in the CAEN project's beginning.

Representing Expert Knowledge

Knowledge engineers represent expert knowledge in a variety of ways. Rule-based expert systems use rules in the form of "if-then" to represent the correct action for a given condition. These are production rules in the sense that given a set of conditions the system predicts the appropriate action. Artificial neural networks represent knowledge through their network topology and connection weights.

One may also divide knowledge into *procedural*, *declarative*, and *tacit* knowledge [Giarratano and Riley, 1994]. Procedural knowledge implies the idea of how to do something. Declarative knowledge concerns information that is known to be true or false. Tacit knowledge involves the ability to do something but not necessarily to know why that something is done. Our reflexes are a good example of tacit knowledge.

Another subdivision involves breaking knowledge into a hierarchy: *meta-knowledge*, *knowledge*, *information*, *data*, and *noise* [Giarratano and Riley, 1994]. Facts tend to be expressed as data or information. Expert systems try to separate data from noise, and may also attempt to process the data into information and then later into knowledge. The meta-knowledge can act upon the processed information, deciding which lower level knowledge is applicable to a particular problem.

Another classical method for representing knowledge is that of a semantic network. Semantic networks can provide the "is-a-kind-of" relations used in reasoning systems. Figure 3 represents a semantic network capturing family relationships. Yet another method of knowledge representation is schemata and frames [Giarratano and Riley, 1994]. Frames do well representing knowledge domains with stereotypical objects.
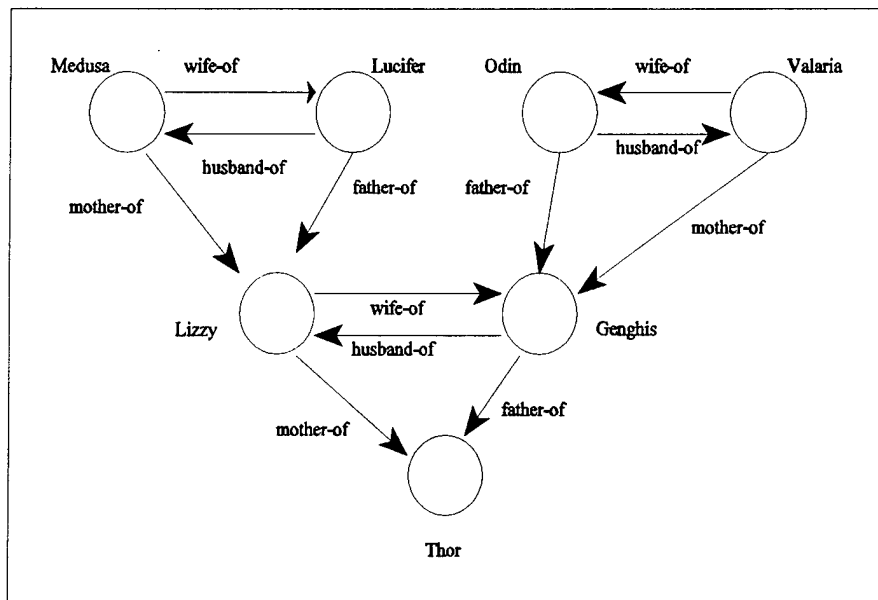
Figure 3: A Semantic Network.

In general, a knowledge engineer must tailor an appropriate knowledge representation model to a specific problem; there is no general solution for all problems.

Uncertainty

"Information pertaining to real-life problems is rarely known with complete confidence: it is usually uncertain" [Henkind and Harrison, 1988]. Exact reasoning applies to a very limited set of domains. Most real life problems use some form of uncertainty in them somewhere. An example is driving in the rain. You know for a fact that water is on the

road, that it does not look deep, and as long as the water is less than an inch deep your tires will hold. You have no idea exactly how deep the water is, but you make the decision of whether to drive on or not anyway. Real life is full of similar problems.

Uncertainty in expert systems can arise from the lack of quality inputs. Most sensors are extremely noisy and therefore provide at times unreliable information. Probabilities may come into play; empirical information may show that a given activity occurs only once in one million tries. Weak implications can also introduce uncertainty. For example, "Given that John is driving a Rolls Royce, one may infer that John is rich, but the conclusion is not completely certain because John may have stolen the car or John may be the chauffeur, etc." [Henkind and Harrison, 1988]. Unclear semantic meanings can also yield uncertainty. What does "cold" translate to in degrees Celsius?

Four prominent methods for representing uncertainty are Bayesian reasoning, Dempster-Shafer theory, fuzzy set theory, and MYCIN/EMYCIN calculus [Henkind and Harrison, 1988].

Bayesian reasoning combines probabilities with information to garner some idea of how certain the information is. The probability of occurrence of a specific fact provides the certainty for that fact. It is thus necessary to gather all the prior probabilities on each pertinent fact. The need to calculate these prior probabilities is one of the drawbacks of this method. Furthermore, the assumptions of independence of evidence and disjoint conclusions restrict its use in some applications.

Dempster-Shafer theory utilizes a set of conclusions which are assumed to be exhaustive for the domain and independent of one another. This theory relates evidence to

subsets of the domain and deals with calculation measures of beliefs and doubts. This method can yield nonintuitive results, and again the assumption that evidence is independent is not always supportable by facts [Henkind and Harrison, 1988].

Fuzzy-set theory handles uncertainty by assigning set memberships to assertions [Henkind and Harrison, 1988]. Fuzzy sets are not necessarily disjoint. The theory of approximate reasoning grew as an extension to Zadeh's work in fuzzy set theory. It applies rules of compositional inference to allow application of fuzzy theory to classical inference strategies and provides low complexity for computing uncertainty.

EMYCIN/MYCIN calculus arose from efforts to support uncertainty in medical diagnoses. "The fundamental idea behind MYCIN calculus is that for each hypothesis certain pieces of evidence tend to confirm it, while others tend to disconfirm it" [Henkind and Harrison, 1988]. The calculus uses measures of belief and disbelief to represent uncertainty and has well-specified rules for combining evidence based on the values of the incoming certainty factors for each piece of evidence. This calculus is linear in complexity but may give nonintuitive results in situations where one must combine a great deal of evidence or chain several inferences together.

These four methods of implementing uncertainty are highly interrelated to one another [Henkind and Harrison, 1988]. In various instances one can reduce one method to another. The choice of a particular method ultimately relies on the problem at hand. Bayesian methods do very well in domains where the probabilities are known or easily attained. Dempster-Shafer applies well when the problem yields uncertain values based on sets rather than individual items. Fuzzy-sets are quite flexible and have low information and time complexity

while MYCIN/EMYCIN offers a simple method for representing and combining evidence [Henkind and Harrison, 1988]. Due to the lack of accessible prior probabilities or set membership values, the CAEN team chose an EMYCIN calculus-based approach for representing and reasoning with uncertainty. Chapter 3 addresses the particulars of this approach.

CHAPTER 3

AUTOMATED GC FAULT DIAGNOSIS

Two main approaches to automating Gas Chromatography (GC) fault diagnosis are rule-based expert systems and expert networks. In the transition from rule-based systems to expert networks, knowledge acquisition plays a key role. Chapter 3 covers the two approaches and knowledge acquisition.

Rule-based Approaches

Lahiri, Stillman, and fellow colleagues tackle the problem of building an expert system from a truth table for determining the analyzability of gas chromatograms [Lahiri and Stillman, 1992]. George Luger and Joel Matek at the University of New Mexico approach the same problem from a straightforward rule-based implementation utilizing Gensym's G2 expert system package [Gensym, 1993].

Using a Truth Table

At the University of Western Ontario, Stillman uses a truth table representation of the relationships between symptoms and causes in GC fault diagnosis. These tables provide a matrix crisply relating a list of symptoms to a given cause and vice versa. Stillman and his students build these tables from their extensive practical experience in the field.

19

In these truth tables, a "T" entry in a cell indicates the designated symptom will appear in the chromatogram when the cause indicated is present. An "F" indicates that a symptom definitely will not appear if this cause occurs. Blanks indicate no relationship between symptom and cause. In an effort to automate the analysis process, they utilize strict rule-based expert systems. They implement these expert systems in a Microsoft Windows environment [Stillman et al., 1991].

Stillman experiences some difficulties in representing qualitative relationships between symptoms and causes. The relationships between causes and symptoms are sometimes complex and a simple true/false relationship does not hold. Stillman tries many different combinations of "if-then" rules to obtain correct predictions.

Stillman subdivides the causes into various categories of severity. For example, he uses *Dirt in the Injector* and *Dirt in the Injector (severe)* [Stillman, 1993]. He does not subset the symptoms into various levels of severity. The lack of symptom subsetting restricts the type of relationships expressed between the symptoms and the causes. Utilizing true/false entries, there is no method to discriminate between varying levels of severity as related to different causes. Additionally, a minute appearance of a symptom, possibly due to an error in the analysis of symptom presence, results in a symptom carrying a value "T" through his system.

Stillman's fit of expert systems to the problem at hand produces for the first time truth tables to represent the chemist's expertise and a method for automating their use. His tables provide the starting point for the CAEN project.

As the knowledge space grew and became more complex, the brittleness of Stillman's "if-then" rules and lack of uncertainty handling ability proved to limit the flexibility of his system. For example, Stillman connects the symptoms *Unresolved peaks* and *Band broadening* to three causes: *Carrier gas flow too low*, *Injector temperature too low*, and *Column degradation*. In a chromatogram showing only these two symptoms, Stillman's system finds difficulty in differentiating among these three causes.

Using a Knowledge Table

In work with Luger at the University of New Mexico, Matek develops a rule-based expert system from an extension to Stillman's table.

Matek uses linguistic qualifiers described in the knowledge acquisition section of this chapter to build a rule base to predict causes. Although Matek provides a more complex system than did Stillman, the confidence factors on his extensive set of rules are heuristically set.

Figure 4 contains an example of Matek's rules. The rules include expressions like (the value of SensitivityChange of GC * 0.75). The values found in these expressions in the same position as 0.75 above are the certainty factors

for any gc-peak GC that is part-of any gc-chromatogram that is generated-by any gc-instrumentation GI
if ((the value of the
SurrogatePrecision of GC * 0.5) +
(the value of the SensitivityChange
of GC * 0.75) + (the value of the
GhostPeaks of GC * 0.5)) >= 0.005
then conclude that the problem-cause
of GI-13 = "Dirt in syringe" and
conclude that the problem-code of
GI-13 = "003" and conclude that
the problem-sub-system of GI-13 =
"syringe" and conclude that the cf
of GI-13 = ((the value of the
SurrogatePrecision of GC * 0.5) +
(the value of the SensitivityChange
of GC * 0.75) + (the value of the
GhostPeaks of GC * 0.5)) * 0.57
and show gc-systems--breakdown-causes and change the arrow icon-color of arrow-syr to red and
conclude that the explanation of
GI-13 = "Since the
SurrogatePrecision = [the value of
the surrogateprecision of GC] and
the SensitivityChange of GC =
[the value of the sensitivitychange
of GC] and the GhostPeaks of GC
= [the value of the ghostpeaks of
GC] then the problem cause is:
Dirt in syringe" and conclude that
the solution of GI-13 = "Clean or
replace syringe"

Figure 4: Example rule in Matek's system.

for each symptom entry found in the knowledge table and Matek sets them using heuristics. The inability to learn the certainty factors limits Matek's system.

Matek employs an excellent interface for his system to the Task Sequence Controller (TSC), develops an extensive system to deal with various configurations, and provides a detailed help facility, all written in G2. The CAEN team will likely incorporate many of the features from Matek's system into their expert networks.

Knowledge Acquisition

Prior to and during the building of the CAEN networks, the Analysis Assessment team for the Data Interpretation Module (DIM) concentrated on building and refining the basic knowledge table. The CAEN team has the responsibility of managing this knowledge acquisition effort. This section covers some background information and the experiences the CAEN team gained from the knowledge acquisition process.

Background Information

"Knowledge acquisition and representation is the knowledge engineering job of acquiring and organizing the knowledge needed to develop a knowledge-based system" [Mockler and Dologite, 1992]. The goal is to capture and represent expert knowledge of a given domain. Knowledge engineers must organize information in a way that allows efficient use in knowledge-based systems. The capture and organization of knowledge can be a very complex and lengthy process.

There are many sources of expert knowledge. Expert knowledge may come from communication with experts, analysis of visual or audio recordings of experts in action, or reviews of knowledge sources (reports, books, etc.).

Knowledge engineers require interpersonal skills for interviews to be effective. It is not uncommon to find that experts feel threatened by the idea that the developed knowledge-based system will replace them.

After establishing a working relationship with the experts, the knowledge engineer attempts to discern the reasoning process used by experts during the interviews. The knowledge engineer can have the expert simply explain how he or she makes certain decisions, or may prompt the experts in a more guided approach. The knowledge engineer must be careful to discriminate between the reasoning process and the facts/beliefs used by the expert. Once one captures the reasoning process, a structured question and answer period can provide the facts necessary for a given problem.

In certain circumstances it may be difficult for a domain expert to describe his or her actions and reasoning process. When no set process is available or apparent, a knowledge engineer can pose a problem to an expert and observe his or her development of a solution. The knowledge engineer can review recordings (tapes, notes) of problem solving sessions later to discern the reasoning process.

Finally, a knowledge engineer can attempt to obtain all the information at one time or can use an iterative/evolving approach. The single meeting approach may work well for limited and well defined domains. An iterative approach is usually best for a long term project or where the domain expertise is sparse and conflicting [Mockler and Dologite, 1992].

The CAEN Team's Knowledge Acquisition Efforts

The CAEN team became the knowledge engineers for the analysis assessment portion of the Data Interpretation Module (DIM) project at a meeting in Albuquerque in May 1994.

At the meeting of analytical chemists, signal processing engineers, and computer scientists, the group utilized Stillman's table as a starting point for automating the fault diagnosis process. The group first discussed and agreed upon the set of possible symptoms and the set of possible causes critical to the diagnosis problem. The knowledge table in Appendix A contains the members of these sets. Fault diagnosis experts then proceeded to debate the relationships between the symptoms and the causes.

The CAEN team guided the experts in their discussion of the interrelationships between the symptoms and causes. With the aid of the CAEN knowledge engineers, the experts quickly realized the limitations of Stillman's true/false approach. The experts were more comfortable describing a particular cause and its related symptom suite using terms describing the frequency and severity of symptoms.

Along with the experts, the CAEN team developed a set of useful linguistic qualifiers to describe the relationships between the symptoms and causes. The proposed set of qualifiers, still in use by the DIM team, is ALWAYS, USUALLY, SOMETIMES, INFREQUENTLY, and NEVER. These qualifiers have dual meanings: they relate a frequency of occurrence of a given symptom to a particular cause and also relate the symptom severity of a given symptom to a particular cause. The CAEN team refers to these as *AUSIN* factors.

To illustrate, the table in Appendix A shows an "A" in the symptom row corresponding to *Sensitivity Change* and the cause column of *Leaking Syringe*. The meaning of this entry is, roughly, "When the cause *Leaking Syringe* occurs, symptom *Sensitivity Change* is always apparent in the gas chromatogram."

Note the manner in which the information in this example relates a cause to a symptom. Given a cause, this symptom appears with a certain frequency. This inferencing direction is opposite to the manner in which the CAEN team must use the information in the automated system. Symptoms will appear in gas chromatograms and the DIM must determine the corresponding cause.

The May 1994 Albuquerque meeting resulted in a partial knowledge table using AUSIN factors. The CAEN team then utilized a series of electronic mail questionnaires to fill in the remaining symptom to cause relationships. Although ultimately effective, the electronic mail questionnaire method progressed at a non-constant rate due to differences in the timeliness of responses. Further meetings and conversations with the experts have and will continue to shape the knowledge table.

From the experts, the CAEN team also determined that the table entries varied depending on machine configuration, machine manufacturer, sample type, and other factors. Frequent changes to table entries prompted research to automate the process of creating CAEN networks from knowledge tables [Levis et al., 1995].

The CAEN Expert Network Approach

The CAEN team proposed use of expert networks to solve the problem of GC fault diagnosis. The CAEN team originally intended to utilize expert networks to refine the rule-based approaches described earlier. The experts' table-based approach to knowledge representation forced the CAEN team to re-examine the traditional use of expert networks. The use of linguistic qualifiers described above also led the CAEN team to develop and implement a new inference mechanism.

Initial Strategy

Working initially with only the Stillman true/false table, the CAEN team decided to compute the certainty for a given conclusion as the percentage of expected symptoms present for a given cause. For example, in a cause with five expected symptoms and only two out of the five symptoms present, conclude this cause with a forty percent certainty. The symptom input values propagate through the entire network, yielding values for causes. The CAEN network chooses the cause with the highest value.

Extension of Expert Network Definition

Mapping a truth, or knowledge, table to a network and vice versa extends the traditional concept of expert networks developed by Kuncicky and fellow colleagues at Florida State University. The original body of work in expert networks focused on a mapping from a rule-based expert system with a associated inference engine. The CAEN team's extension precludes the need to implement a rule-based expert system prior to using expert network technology. Information in table form can be mapped to a rule base representation of the form a $\rightarrow$ b (cf) where a is a symptom, b is a cause, and cf is a value associated with a table entry. With this inherent mapping, the CAEN team creates their expert networks directly from the tables.

Development of a New Inference Mechanism

The experts' use of AUSIN factors generated the need for a new inference mechanism. The CAEN team debated at length on the information contained in each AUSIN factor. Did an AUSIN factor relate the effect of symptom severity to a particular cause? The relationship to severity implies that a symptom with an ALWAYS entry in the table should have greater

contribution to a conclusion than a symptom with a SOMETIMES entry. Or does a knowledge table entry relate the frequency of occurrence of a symptom to a cause? The frequency relationship implies that a symptom with an ALWAYS entry must appear if the network is to choose this cause. If an ALWAYS symptom does not appear, the network should discount this cause in some manner. The CAEN team found these issues difficult to resolve. The CAEN team was unable to extract the experts' intended meaning for the table entries even after multiple interviews and conversations with the experts.

Implementation of the New Inference Mechanism

The resolution of the AUSIN interpretation problem is represented by the CAEN team's implementation of the GC fault diagnosis system as a four layer hybrid network in G2. The team's solution also supplies a mapping from the knowledge table to the network and an inference algorithm. Appendix D shows the portion of a typical CAEN network for two causes, *Column Bleed* and *Column Degradation*, and illustrates the four layers of the network.

Layer zero provides *Symptom* nodes that represent symptom inputs. Layer zero nodes connect to one or more nodes in layer one. Layer one has *Filter* nodes that map directly to the AUSIN factors. Filter nodes receive input from only one Symptom node and connect to only one node in layer two. Layer two has *Combination* nodes that collect evidence from one or more Filter nodes. Combination nodes connect to an output node in layer three. Layer three contains *Cause* nodes that report the final prediction values of the network.

Layer Zero. The CAEN network receives values for the Symptom nodes from a symptom file illustrated in Appendix B. This input file contains symptom values provided by

signal processing routines that determine symptom presence in gas chromatograms and is readable in G2. The output connection of each Symptom node has a fixed weight of 1.0. The possible values for these Symptom nodes are in Table 1.

Table 1

Symptom Values and their Meanings

Range/Value	Meaning
From 0.00 to 1.00	Severity of symptom in GC data, 1.00 is a maximum
0.00	Symptom is not present in the chromatogram
-1.00	Impossible for symptom to occur in this sample
-2.00	Did not check for this symptom in the sample

The -1.00 and -2.00 are special flag values developed jointly by the CAEN team and the signal processing team. These flag values capture critical expert knowledge that would otherwise be reported as 0.00.

A symptom receives the -1.00 value when the specific machine configuration or mode negates the possibility of this particular symptom appearing in a chromatogram. For example, in an isothermal mode the symptom *Baseline Drift Rising* cannot occur (See Appendix A).

The signal processing engineers assign the -2.00 flag value when their routines can not evaluate a particular symptom in a chromatogram. The inability to evaluate may be due to the hierarchical structure of the signal processing routines. The presence of one symptom or a particular sample type may preclude, through data masking, the ability to discern other symptoms in the chromatogram.

For example, experts only look for *Ghost Peaks* during blank samples that have no contaminant. This blank sample should not produce any symptoms. If symptoms appear,

e.g., *Ghost Peaks*, there is a fault in the equipment. Experts cannot distinguish *Ghost Peaks* from true peaks in a normal sample.

Layer One. Layer one has Filter nodes with different functionality for each linguistic qualifier in the table entries. Each Filter node has the following attributes: *threshold, Good-Dog Factor, Bad-Dog Factor*, and *output connection weight*. Table 2 displays default values for each node type.

Table 2

Filter Node Values in CAEN

Node Type	Threshold	Good-Dog	Bad-Dog	Weight
ALWAYS	0.1	1.0	-0.5	0.85
USUALLY	0.1	1.0	-0.25	0.75
SOMETIMES	0.1	1.0	-0.01	0.50
INFREQUENTLY	0.1	1.0	-0.001	0.25
NEVER	0.1	-1.0	0.0	0.85

The threshold relates to the symptom severity and acts as a flag to the firing function of the node. If the symptom input value exceeds the threshold, the Filter node fires differently than when the symptom value is below the threshold. The Good-Dog factor relates a measure of reward to the symptom input based on its AUSIN factor table entry. The inference mechanism in CAEN utilizes the Bad-Dog factor when the given input falls below the threshold. The use of the Bad-Dog factor relates a measure of penalty for a symptom not appearing when expected. The default output connection weights address the importance of a given table entry to a cause.

Note that the Good-Dog and Bad-Dog values for NEVER nodes serve to reward symptoms not appearing and punish those that do appear. Define u as the activation value of a Filter node in the network. The firing function of a Filter node is as follows:

$$u = \begin{cases} (\textit{input-value} * \textit{input-weight}) * \textit{Good-Dog}, \\ \quad \textit{if (input-value} * \textit{input-weight)} > \textit{threshold} \\ ((1.00 - \textit{input-value}) * \textit{input-weight}) * \textit{Bad-Dog}, \\ \quad \textit{otherwise.} \end{cases}$$

A value of 0.00 for u is assigned when the special flag value of -1.00 or -2.00 is received from a connected Symptom node. The value u is propagated along the node's output connection.

Layer Two. Combination nodes in layer two relate to an entire column in the knowledge table. Each Combination node has an input connection for each AUSIN entry in the knowledge table. The Combination nodes combine all the incoming evidence in an EMYCIN-like fashion.

Let P be the set of input connection indices from Filter nodes whose activation value u times the connection weight w is positive, that is, $P = \{i \mid u_i w_i > 0.00\}$. Let N be the set of input connection indices from Filter nodes whose activation value u times the connection weight w is zero or negative. Following EMYCIN, define two equations:

$$y^+ = 1 - \prod_{i \in P} (1 - u_i w_i)$$

and

$$y^- = -1 + \prod_{i \in N} (1 + u_i w_i) .$$

The Combination node's output, z, is

$$z = y^+ + y^-.$$

The z value flows along the output connection from layer two to layer three.

Layer Three. Cause nodes in layer three receive their values by multiplying the incoming value and weight on their input connections from Combination nodes. Currently, there is only one Combination node connected to each Cause node.

Parallel Paths. Multiple Combination nodes leading to a single Cause node indicate the presence of parallel paths where two (or more) different subsets of the entire symptom set lead to the conclusion of the same cause. These parallel paths may appear as different cause columns in the table with different sets of symptoms leading to the inference of the same cause. The common Cause node, representing these different columns in the knowledge table, will choose the maximum value from among the parallel paths.

Training. The training algorithm now in use alters the weights, w_i, between layers one and two. Chapter 5 describes this training algorithm. Future work in the CAEN project will address training other weights and factors in the networks.

Conclusion

Early attempts to automate the data interpretation process in gas chromatography laid the groundwork for the CAEN project. Currently, within the analysis assessment project of the DIM there are two approaches implemented: the rule-based system approach and the expert network approach, both using the AUSIN linguistic qualifiers. The knowledge

acquisition process used to build the knowledge tables was critical to both projects. The

CAEN team implemented a new inference mechanism for the knowledge table.

CHAPTER 4

NETWORK ARCHITECTURE ALTERATION TECHNIQUES

Introduction

Network-based systems have many advantages over rule-based systems, but one of the drawbacks of network systems is having to determine the architecture to build into the network. Early on in the Contaminant Analysis Expert Network (CAEN) project, the team recognized the need for an automated method to improve the architectural structure of their networks. This chapter contains a review of the four prominent algorithms from the network architecture alteration literature considered by the author in his research: Cascade Correlation [Fahlman and Lebiere, 1990], Stack [Fang and Lacher, 1993], Optimal Brain Damage [Cunn, Denker, and Solla, 1990], and Optimal Brain Surgeon [Hassibi, Stork, and Wolff, 1992]. Chapter 6 contains a discussion of the network alteration method developed as part of the CAEN project.

There are two basic strategies employed in the automated network alteration methods: a *constructive*, or growing, approach; and a *destructive*, or shrinking, approach. Two of the algorithms, Cascade Correlation and Stack, are constructive algorithms. In a constructive algorithm, the method begins with a small network, trains the network, and computes total network error. If the network error is not within tolerance, additional nodes or layers are

added to the network and the process of training and testing continues. A constructive method continues until the algorithm constructs a network that produces the desired results.

Optimal Brain Damage and Optimal Brain Surgeon are destructive algorithms. In a destructive algorithm, the algorithm begins with a larger than necessary network, trains this network, and then tests it. If the results are acceptable, the algorithm searches the network for unneeded connections and removes them. The overall goal of destructive algorithms is to prune the network to the smallest configuration that still produces acceptable results.

Cascade Correlation

Fahlman and Lebiere of Carnegie-Mellon University developed the Cascade Correlation Algorithm [Fahlman and Lebiere, 1990]. The algorithm begins with a fully connected one-layer network. A fully-connected network is one in which each node in a layer connects to each node in the next layer. One-layer implies the existence of input nodes and output nodes only. The input layer includes a bias node, normally with a +1 value. Figure 5 illustrates a typical fully connected network.
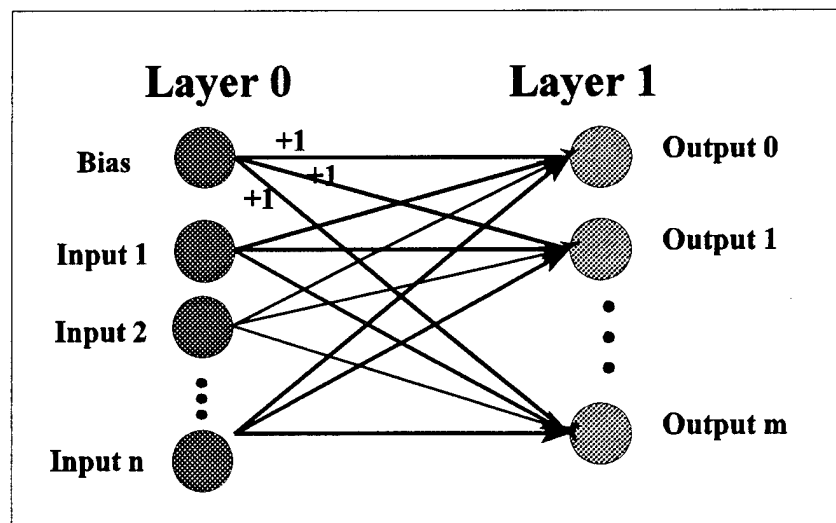
Figure 5: A Fully Connected One-Layer Network.

Training occurs on the network for a specific amount of time represented by a *patience parameter* [Fahlman and Lebiere, 1990]. If the error is within the specific range required, training is complete. If the error still falls outside the accepted range, the method creates a *candidate node* [Fahlman and Lebiere, 1990]. Cascade Correlation fully connects the candidate node to the input layer. In addition, the method fully connects the candidate node to any previously added candidate (now called a hidden node). See Figure 6.
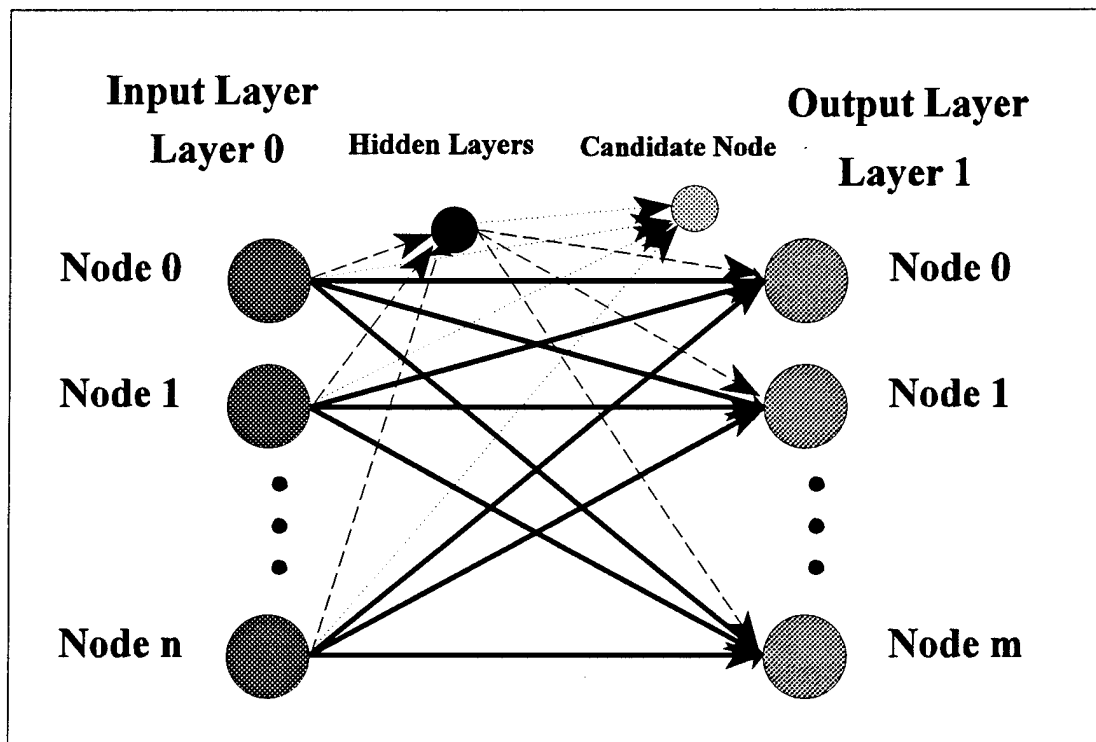
Figure 6: Adding a Candidate Node in Cascade Correlation.

Cascade Correlation tunes the candidate node's weights on its incoming connections by training. The method then fully connects the candidate node to the output layer and trains the entire network seeking to minimize network error. This incrementally constructive process continues until the network achieves the desired result [Fahlman and Lebiere, 1990].

The most important contribution of Cascade Correlation is that the network determines its own size. The automatic size determination normally provides a network that is not oversized [Fahlman and Lebiere, 1990]. The main drawback encountered when the author considered the Cascade Correlation approach for this research is the loss of knowledge. Once Cascade Correlation adds additional nodes, the clarity of relationships between symptoms and causes disappears.

Stack Algorithm

Fang and Lacher at Florida State University proposed the Stack learning algorithm [Fang and Lacher, 1993]. The Stack algorithm utilizes the perceptron learning algorithm. Again, the algorithm begins with a one-layer fully connected network with a bias node. The bias node has value +1 (See Figure 5). The network receives training until the error change stabilizes. If the error change is within acceptable limits, the process terminates.
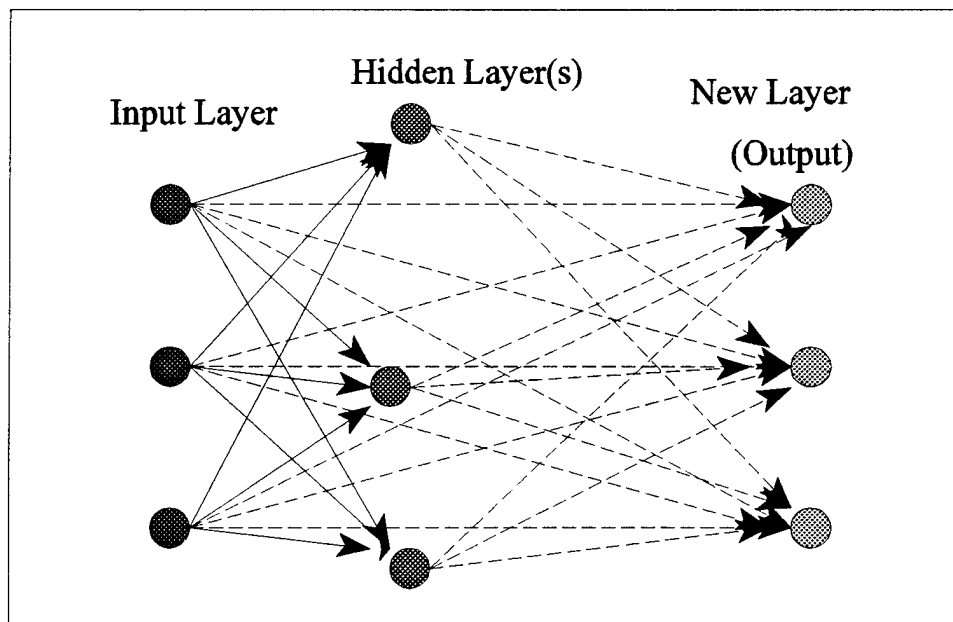
Figure 7: Adding a New Layer with Stack.

If the error is still out of the desired range, an entire new layer of nodes is added as in Figure 7. This added layer becomes a new output layer. The existing output layer becomes a hidden layer. The Stack Algorithm then connects all nodes in the previous network to the new output layer. These connections receive random weights in the range of 0.00 to 1.00.

When training resumes, the new weights undergo training. The Stack algorithm iterates until it produces a network that reaches the desired error range. The Stack algorithm provides efficient training for dynamic network topologies, making the algorithm attractive for unknown domains [Fang and Lacher, 1993].

Again, the drawback with the Stack approach is the loss of relational knowledge. As the layers increase there is no effective way to rationalize what the new connections and their associated weights mean.

Optimal Brain Damage

Developed by Cunn, Denker, and Solla of AT&T Bell Laboratories, Optimal Brain Damage (OBD) is a destructive algorithm [Cunn, Denker, and Solla, 1990]. The OBD algorithm is effective in yielding a faster and more accurate network, although it suffers from many of the same problems encountered by constructive algorithms.

OBD begins with a fully-connected network which is larger than necessary in size. The algorithm removes connections in the network based on contribution to network error and retrains the network. OBD iterates the destruction and retraining cycle until the network exceeds a given error tolerance. OBD keeps the last network that remained within the tolerance [Cunn, Denker, and Solla, 1990].

The destruction of connections bears no direct relationship to the dependency between a given symptom and cause. Changes to the network structure result in the loss of knowledge contained in the network topology.

Optimal Brain Surgeon

Hassibi of Stanford University along with Stork and Wolff of the Ricoh California Research Center developed the Optimal Brain Surgeon (OBS) algorithm [Hassibi, Stork, and Wolff, 1992]. As in OBD, OBS begins with an overlarge network which must successfully train to an acceptable error. The OBS algorithm then compares the connection *saliency* (a relation to contribution of error) to the network error. If the saliency factor compared to the network error is small, the algorithm deletes the connection. OBS removes a connection and retrains the network. OBS terminates in the same manner as OBD.

Although the OBS algorithm outperformed OBD in several test cases, OBS has the same difficulties as OBD [Hassibi, Stork, and Wolff, 1992]. Once again, the method of removing connections alters the relationships between symptoms and causes in a manner that disallows recovery of knowledge contained in the network's connection topology.

Conclusion

The literature contains many methods for automatic network alteration to improve performance. Researchers have effectively used the Cascade Correlation and Stack algorithms to increase the complexity of weight space and provide a better fit of a network to a problem domain. Optimal Brain Damage and Optimal Brain Surgeon provide a means for trimming large networks of unnecessary connections. All of these approaches lack the

ability to relate these alterations to physical relationships between inputs and outputs. These algorithms improve network performance but destroy the user's ability to recover knowledge from the network into a knowledge table.

The Data Interpretation Module project dictated the need to build networks that realistically mirror the knowledge structure used by human experts. The CAEN project's requirement of retaining symptom/cause relationships in some form when altering their networks' structure provides the base motivation for the research described in Chapter 6.

CHAPTER 5

RESEARCH TOOLS

Introduction

There are four main research tools utilized in conjunction with NetMedic: G2, a Gensym product; the Contaminant Analysis Expert Network (CAEN) network itself, a module in G2; a G2 module that creates a CAEN network from a common file format; and a standard back-propagation training program written in C. All four research tools are integral to NetMedic's use.

Gensym

G2 is a commercial object oriented industrial control software package developed by Gensym for fully automated plant/process control [Gensym, 1993]. With its built-in expert system capabilities, G2 can inference with either forward or backward chaining. The object oriented approach of G2 allows easy representation of real world items as objects with associated properties, attributes, and relations.

The structure of the language in G2 allows a very flexible means of accessing objects by name, location, type, with value, etc. G2 organizes objects on workspaces. These workspaces are themselves objects, so the developer may dynamically create, alter, and delete workspaces and the objects upon them.

The developer operates on objects using rules, functions, and procedures. Rules are typically in the "if-then-else" format, and can be set to cause backward chaining or to fire on a periodic basis. The user may also set priority levels among rules to resolve conflicts.

Functions in G2 return a single value when called. Procedures provide the most flexibility. The procedural language is similar to C and is self-prompting in the syntax category. Basic conditional and loop programming constructs are available.

G2 also provides a rich set of predefined data types, both generic objects and generic connections. The developer can manipulate objects by writing code to operate on connected objects, for example, "Turn every object X connected to object Y green."

The object oriented G2 language provides a useful interface for the development of CAEN networks.

CAEN Module Overview

The CAEN network is not a stand alone application within the Data Interpretation Module (DIM). The overall goal of the DIM project is to field automated Gas Chromatography (GC) data interpretation software for use with a Task Sequence Controller (TSC) in an automated lab. The Department of Energy's current plan is to have the automated lab completely contained within a semi-trailer, including all equipment and integrated software.

The TSC starts the CAEN application by selecting a particular GC machine configuration. The CAEN module loads the proper expert network file, based on the machine configuration. In the absence of an expert network file particular to the configuration requested, the CAEN module uses a generic base network file.

If the network used by the CAEN module does not meet performance specifications, the user calls NetMedic to analyze the data and suggest structural modifications to the network. The structurally modified network receives training. The CAEN module evaluates the performance of the trained network. If the performance of the new network begins to deteriorate, the CAEN module invokes NetMedic again and the process repeats. To develop historical GC training data online, the CAEN module requires user input if a fault occurs.

Once the CAEN module meets performance criteria and is deployed online, it processes a symptom file to determine whether the data is analyzable or not. If a fault is detected, the CAEN module advises the TSC of the cause and the suggested remedy. The TSC then either automatically replaces/fixes the problem or notifies the operator to do so. If the CAEN module does not detect a fault, the TSC passes the gas chromatogram to another module within DIM for analysis of contaminants contained in the sample represented.

NetMaker

As noted earlier, the knowledge tables used in the CAEN project changed frequently as chemists debated back and forth about symptom/cause relationships. The CAEN team realized the need for different knowledge tables based upon differing machine configurations. Alan Levis headed the effort to automate the creation and storage of G2 expert networks from text based knowledge tables such as those in Appendices A, I, and J. The automated tool is called *NetMaker*.

In order to automate use of the knowledge tables, the CAEN team designed a common network file format to represent a knowledge table or an associated CAEN network.

Appendix E contains an example `common.net` file. Figure 8 shows the format for the

`common.net` file.

```
Section 1 Comments
# A line with a preceding '#' indicates a comment

Section 2 Symptom to Combination Node Information
Workspace-Name Symptom-name Combination-node-number
Symptom-to-Filter-Weight Filter-node-threshold
Good-Dog-factor Bad-Dog-factor
Filter-to-combination-node-weight

Section 3 Combination to Cause node Information
Workspace-name Combination-node-number Cause-name
Combination-to-cause-weight
```

Figure 8: **File Format for `common.net`**.

The `common.net` file has three main sections as shown. In Section 2, Workspace-name indicates a logical grouping of causes by their location on Gas Chromatography (GC) equipment. The Combination-node-number is an internal distinction to identify parallel paths to a specific cause. Section 3 entries relate information for Combination to Cause nodes.

NetMaker takes a common network file and generates the expert network in G2. Upon demand, NetMaker also translates the G2 CAEN networks into the common network file format. NetMaker provides a central point of information exchange between the knowledge tables created by the chemists, the G2 expert networks, NetMedic, and the training algorithm used in the CAEN project.

Backpropagation

The training package uses a customized backpropagation of error routine. The current training alters only the weights between the Filter and Combination nodes in the CAEN networks. Future work involves expansion of training to other weights and factors

in the CAEN networks. Hruska's C program training algorithm takes as input the common network file and all the training examples. The training attempts to correctly predict the cause at fault and achieve maximal separation between the top predicted cause and the remaining causes. The training to maximize separation is an effort to increase the accuracy of predictions when symptom sets overlap.

CHAPTER 6

NETMEDIC

Overview

In addition to the author's role as a Contaminant Analysis Expert Network (CAEN) team member in designing and building the CAEN expert network, a major individual contribution of the author is the NetMedic tool. NetMedic automates the process of structural alteration of CAEN network and aids in the knowledge acquisition process. NetMedic works by confirming, refining, and augmenting expert knowledge based on statistical analysis of the Gas Chromatography (GC) symptom data.

Experts can use NetMedic in an interactive mode with the expert accepting, rejecting, or deleting connections during consultation with NetMedic. In an automatic mode, NetMedic attempts to best fit a network to the data by implementing all connection proposals. Use of NetMedic improves the CAEN networks' prediction capability based on current data. The prediction problems encountered with the initial CAEN networks (Chapter 7) along with the evolving nature of the expert knowledge provided the necessary motivation for NetMedic.

This chapter will focus on the interactive mode of NetMedic; the automatic mode is merely default acceptance of all NetMedic's proposed changes to the network.

Gaining Statistics

NetMedic begins analyzing data from the symptoms and outputs generated by the CAEN networks. NetMedic develops statistics on how symptom values relate to causes from files produced by the CAEN networks. Several statistics on each symptom are developed in groupings to discern patterns of network performance.

Input Files

In addition to symptom values and machine configuration information, the CAEN network training files have an induced (or expected) cause field to allow supervised learning. This induced cause has a measure of severity field attached which is currently unused due to lack of reliable data.

NetMedic uses output from the CAEN networks including all the symptoms and their values, all the causes and their predicted values, the expected cause, and finally the predicted cause (Appendix C).

Groupings

NetMedic queries the user for the number and location of files he or she wishes to process. NetMedic reads and groups the collected information by expected cause. NetMedic collects the statistical information in three distinct groups for each cause for comparison.

First, NetMedic considers the total set of files in which a particular cause is the expected cause. Second, the same statistics are collected for the set of files where the CAEN network predicted the cause correctly. Finally, NetMedic gathers the statistics on the set of training files which the CAEN network predicted incorrectly. NetMedic labels these

groupings total, good, and bad to provide insight to the performance of the CAEN network with various inputs. See Appendix F.

Statistics Gathered Per Group

NetMedic calculates the mean and standard deviation of symptom and cause values. NetMedic tracks the number of times that a symptom has a 0.00 value, a positive value, a value of -1.00, and a value of -2.00. The total number of times the symptom is non-zero compared to the total number of times a cause occurred gives a measure of relative frequency.

Identifying Connections

Once NetMedic gathers all its statistics, NetMedic prompts the user to select a cause to modify. NetMedic reviews the data for this cause and determines the proposed connection type for each symptom. Currently NetMedic proposes a connection for all symptoms to a cause except those with all -1.00 or -2.00 values.

Determining Connection Type

NetMedic proposes a Filter node type for a connection by analyzing the frequency of occurrence for each symptom in the presence of the cause being considered over the total set of training files. Table 3 illustrates the frequency ranges used.

Table 3

Filter Node Type Frequencies

Filter Node Type	Frequency of Occurrence
ALWAYS	100.00%
USUALLY	90.00% to 99.99%
SOMETIMES	40.00% to 89.99%
INFREQUENTLY	0.01% to 39.99%
NEVER	0.00%

NetMedic also proposes a new threshold for each Filter node. NetMedic uses the mean for the symptom less one standard deviation to determine the proposed Filter node threshold. If the computed threshold is greater than 0.20, the proposed threshold is heuristically set to this computed threshold value less 0.10. The additional subtraction of 0.10 provides a greater range of accepted symptom values and a margin of safety. If the computed threshold value is less than 0.20, the proposed threshold is set to 0.00.

Interacting With an Expert

Once NetMedic completes development of proposed connections for each symptom related to a cause, NetMedic checks for existing connections in the common network file. NetMedic compares its proposed connections to existing connections in the common network file.

Confirming. NetMedic confirms expert knowledge if the existing and proposed Filter node types match and informs the expert of the confirmation. For confirmed connections, NetMedic compares the thresholds and again, if a match is found, NetMedic confirms the expert's knowledge. Otherwise, NetMedic prompts the expert to decide whether to change the threshold as proposed or not.

In addition, NetMedic checks the weight on the current connection. If the connection weight is close to zero, NetMedic queries the expert for a decision to delete the connection. The near zero weight value indicates the lack of importance for that weight in the network in the decision process.

Refining. If the existing and proposed Filter node types do not match, NetMedic attempts to refine the expert's knowledge. It presents the existing and proposed connections

and the statistical data backing the proposal. NetMedic queries the expert to decide what to do. The expert may elect to do nothing, delete the existing connection, or change to the proposed connection. If the expert accepts the new connection, NetMedic prompts the expert about the threshold as explained above.

Augmenting. Once NetMedic processes all the existing symptom connections, NetMedic moves to proposed connections for the remaining symptoms. These connections question the entries that are Blank in the existing knowledge table. The proposal of connections where Blanks appeared in the knowledge table is the augmentation process.

The augmentation process identifies relationships in the data that may be unknown to the expert or simply artifacts of the data. As reported in Chapter 7, the augmentation of expert knowledge did occur several times during use of NetMedic in the knowledge acquisition process. During augmentation, the expert can either add the proposed connection to the network or disregard NetMedic's advice.

NetMedic records the interaction with the expert in a file illustrated in Appendix G.

Removing Connections

Currently NetMedic proposes connections for removal which have small connection weights. It is difficult to distinguish a *no-connection*, or Blank entry in the knowledge table, from a NEVER connection using the statistics calculated in NetMedic. NetMedic proposes a NEVER for symptoms appearing with 0.00% frequency. Experience in using NetMedic indicates experts typically view these connections as blank entries (rather than NEVERs) in the knowledge table.

Automatic Mode Capability

NetMedic has an automatic mode in which it derives connections from data and, without consultation with the expert, builds a network from these connections. The derivation from data only is a valuable technique in the event no expert or expert knowledge is available. It evaluates all the causes at once and makes a new common network file.

However, the automatic mode can create a network that is overspecific to the particular data set used by NetMedic. Although this mode allows the data to determine the structure of the network in the absence of an expert, the use of the automatic mode makes questionable the ability to retrieve expert knowledge from the network.

Appendix H contains an overview of the NetMedic software.

CHAPTER 7

EXPERIMENTAL RESULTS

Overview

The experimental results from this research are reported in three main sections. The first section reports the results achieved by the Contaminant Analysis Expert Network (CAEN) networks based solely on expert knowledge, prior to the creation of NetMedic. The second section contains performance data on NetMedic during its first use with real Gas Chromatography (GC) data. The last section records results of a second pass using NetMedic on new sets of data derived from modified signal processing routines.

Before NetMedic

During the summer of 1994 the CAEN fielded its first expert networks from knowledge tables. Prior to the availability of real training data, the CAEN team developed and tested their networks on simulated ideal data. Once real data was available, the need for NetMedic became clear.

Ideal Data

The initial CAEN network implementation served as a vehicle for testing the intuitive validity of the inference mechanism. Training algorithms used simulated ideal data during this

period. The prototypical trained networks converged and predicted perfectly with simulated data[2] and up to 25% noise.

Real Data

In February 1995, the CAEN project began receiving actual symptom data via Varian Equipment Corporation. The signal processing engineers at LANL took data collected by John Robinson of Varian Equipment Corporation from tests involving intentionally induced faults in gas chromatography instruments. The signal processing engineers extracted a symptom value for each symptom in a range of 0.00 to 1.00. The signal processing engineers based these values on the degree of presence of a given symptom in a particular gas chromatogram. The result of the signal processing routines are symptom files, readable in G2. These files include a field indicating the *induced cause* which allows the CAEN team to verify its results.

The initial data set from Varian represented only four causes. When the CAEN network processed these files based on the entire table (containing all causes), the results were poor. There was heavy competition among causes not represented in the data set. The competition among causes indicated a significant amount of symptom set overlap in the table, of which the CAEN team was already aware.

The original CAEN network that performed perfectly with ideal simulated data managed only 23 correct out of the 103 files of real data.

---

[2] The ideal data simulated for testing had all expected symptoms appearing with values of 1.00 and all other symptoms with values of 0.00 as appropriate for each cause.

NetMedic's First Pass

In its initial use with real data, NetMedic led to the identification of several problems in the data and the knowledge tables used in the expert networks. The CAEN team used NetMedic with different restraints to explore the tool's usefulness in developing networks. A discussion with the experts of NetMedic's results on this first pass was illuminating.

Points of Interest

Symptom always 0.00. The most common problem identified through the use of NetMedic was with symptoms that were always at 0.00 in value for a given cause. If these symptoms appeared with non-blank/non-NEVER entries in the knowledge table, the data conflicted directly with the expert knowledge. This implied either a mistake by the expert, a signal processing problem in detecting the symptom, a configuration specific problem, or a machine specific problem.

Robinson discounted the machine specific error and indicated that these symptoms appeared in actual chromatograms he had analyzed. This left the team to investigate the signal processing routines.

Signal Processing Routines. The goal of the signal processing engineers was to develop highly accurate algorithms that reported a value when the symptom was indeed present and a zero otherwise. In March 1995, the experts concluded that many of NetMedic's proposed connections identified problems in the signal processing routines. The signal processing team revisited their algorithms with added guidance from the GC diagnosis experts. Thus, NetMedic refined expert knowledge by providing feedback to the signal processing engineers.

Symptom ALWAYS has value. Another interesting result was for those symptoms that always had values for the expected cause. The key word is "always," as in "this symptom always appears" with regard to frequency. Thus, when a symptom always had a non-zero value, NetMedic proposed an ALWAYS type connection. This brought into question again the experts' interpretation of the AUSIN factors. Are these factors based upon frequency of occurrence or upon severity of symptom presence?

Frequencies and Rankings. Some symptoms appeared in the real data with different frequencies of non-zero values. The author used an analysis of these frequencies to heuristically determine proposed Filter node types expressed earlier in Table 3. Many of the CAEN networks' predictions for cause values were very close among different causes due to a high number of shared symptoms among the causes. Upon inspection, the close rankings of the predicted causes confirmed the need for NetMedic to manipulate thresholds and use NEVER nodes to increase the discrimination capability of the CAEN networks.

Small Data Set. The initial data set provided symptom data for only four causes. The original CAEN network used before NetMedic had connections for every cause found in the knowledge table. NetMedic could not analyze the other causes to confirm, refine, and augment their connections. The CAEN team reduced the network to only the knowledge table entries for the four causes found in the data set. The modified CAEN network, representing only expert knowledge, evaluated 103 files and predicted 37 out of 103 correctly.

Each cause in the real data shared three symptoms with similar connection types from the knowledge table. Thus, three of the symptom inputs provided information to every cause.

Some of these causes had only one other discriminating symptom. These similarities resulted in the CAEN network picking the wrong cause by a very slim margin. In addition, some symptoms that could aid in discernment were not available due to a lack of signal processing routines.

Parallel Paths. The CAEN team had split some of the cause columns in the knowledge table into parallel paths. These parallel paths shared a common set of symptom connections yet had at least one distinctly different symptom. The CAEN team believed these distinctly different symptoms could not appear in the same chromatogram, and so represented parallel paths to the goal cause.

For example, consider *Leaking Syringe* in the knowledge table in Appendix A. Both *No Peaks* and *Ghost Peaks* are related to this cause in the table. Peaks which appear unexpectedly as ghosts from a previous sample GC are reported as *Ghost Peaks* while *No Peaks* indicates a flat baseline. It would not seem possible for both symptoms to appear and contribute to the conclusion *Leaking Syringe* at the same time. NetMedic found apparent contradictions in the table and/or the symptom processing routines by showing conflicting symptoms appearing at the same time in the data.

Note, this evidence does not discount the idea of parallel cases. The signal processing engineers explained that certain symptoms appearing in a chromatogram mask out the appearance of other symptoms related to the same causes and normally occurred in cases with extreme cause severity only.

The CAEN team removed the parallel paths as appropriate from the network and processed the 103 files again. Now, after reducing the network to represent only four causes and removal of parallel paths, the experts table yielded 54 out 103 correct.

NetMedic's First Pass Results

At the experts' prompting, NetMedic initially treated symptoms that always had a zero value as a no-connection. Utilizing NetMedic in the automatic mode, taking every suggested modification to a connection or threshold, the resultant network predicted 74 out of 103 correctly. The network derived solely from data, with every suggestion (except NEVERs) accepted, did better than the experts.

Different Modes. To determine the impact of NEVER connections, the author tested NetMedic by proposing NEVER connections where the symptom value was always 0.00 as described in the previous chapter. NetMedic reanalyzed the data and produced a different CAEN network that predicted 84 of 103 causes correctly. Without training, NetMedic achieved nearly 80% accuracy. With training, NetMedic's best network predicted 94 out of 103 correct.

Results of Dialogue with the Experts. During a project meeting in March 1995 in Tallahassee, the CAEN team, analytical chemists, and the signal processing team discussed NetMedic's results. The CAEN team compiled the table in Appendix I with only expert knowledge received prior to the Tallahassee meeting. The table in Appendix J is the table after the Tallahassee meeting. Only four causes had actual sample data collected and not all the symptoms found in the knowledge table had signal processing routines tailored for them. The tables below illustrate the results of the dialogue with the experts. In each table, existing

connections are indicated in the second column, NetMedic's proposed connections are shown

in the third column, and the final decision by the experts is recorded in the fourth column.

Comparison of these three columns indicates NetMedic's influence on the makeup of the

knowledge table.

Table 4

Key NetMedic Findings for Column Degradation

Symptom	TABLE	NET-MEDIC	FINAL	ACTION	
RetentionTimeChange	A	A	A	Confirm	
PeakTailing	S	U	S	Confirm	connection
LeadingPeaks	S	A	S	Confirm	connection
SensitivityLoss	None	A	None	**Augment**	Signal Processing error
IncreasingBaseline	None	S	S	**Augment**	New Knowledge

Table 4 illustrates where NetMedic detected connections that did not exist according

to expert knowledge alone. The signal processing engineers recognized a mistake in their

routines. NetMedic provided feedback to the experts, in this case the signal processing

engineers, to augment their knowledge.

Table 5

Key NetMedic Findings for Column Bleed

Symptom	TABLE	NET-MEDIC	FINAL	ACTION	
RetentionTimeChange	N	A	I	**Refine**	Move toward proposal
PeakTailing	None	U	None	**Augment**	Signal Processing error
LeadingPeaks	None	A	None	**Augment**	Signal Processing error
Sensitivity Loss	N	A	None	**Augment**	Signal Processing error
IncreasingBaseline	A	U	A	Confirm	connection
Nonzerobaseline	S	S	S	Confirm	knowledge

Table 5 identifies a key refinement feature of NetMedic. The proposal of an

ALWAYS connection for *Retention Time Change* prompted the experts to rethink their

NEVER connection, finally settling on INFREQUENTLY for the connection. NetMedic

caused the experts to move closer to the proposed connection and indicates a refinement.

Table 6

Key NetMedic Findings for Leaking Septum

Symptom	TABLE	NET-MEDIC	FINAL	ACTION
RetentionTimeChange	S	A	S	Confirm connection
Peak Tailing	N	U	N	**Augment** Signal Processing error
UnresolvedPeaks	S	None	I	**Refine** Experts move near proposal
LeadingPeaks	None	A	None	**Augment** Signal Processing error
SensitivityChange	A	A	A	Confirm knowledge
ReplicatePrecision	None	S	S	**Augment** New Knowledge

Table 6 illustrates the discovery capability of the NetMedic. With *Replicate Precision*, NetMedic's suggestion prompted the experts to alter the knowledge table.

Table 7

Key NetMedic Findings for Leaking Syringe

Symptom	TABLE	NET-MEDIC	FINAL	ACTION
RetentionTimeChange	None	A	None	**Augment** Signal Processing error
Peak Tailing	None	A	None	**Augment** Signal Processing error
LeadingPeaks	None	A	None	**Augment** Signal Processing error
SensitivityChange	A	A	A	Confirm knowledge
IncreasingBaseline	None	A	None	**Augment** Signal Processing error
ReplicatePrecision	A	N	A	**Augment** Signal Processing error

Table 7 illustrates the further usefulness of NetMedic. The signal processing engineers reviewed NetMedic's proposals and statistics and determined that their signal processing algorithms required further work. The feedback to the signal processing team proved extremely useful to the overall CAEN effort.

Review of First Pass Results

The usefulness of NetMedic is clear. The new table created after the first pass of NetMedic predicted 27 out of 84 samples correctly (the CAEN team reduced the set of data files (no longer 103) by removing control, blank, and other inappropriate files).

Impact of Nevers. The experts declined to accept most of NetMedic's suggested NEVER connections. The experts believed there was a no-connection or Blank entry in the

table. The author then modified the new table to add the NEVER nodes omitted by the experts in the tables above. The addition of the NEVER nodes resulted in a network that predicted 40 out of 84. The increased prediction performance through the addition of NEVERs questions the Blanks in the knowledge table, seeming to indicate that there is an advantage to considering a Blank in the table as a NEVER rather than a no-connection during inferencing.

Thresholds. The adjustment of thresholds in effect fine tunes the network and tends to improve performance. The signal processing engineers utilize a threshold in reporting symptom strengths to remove noise from their input chromatograms. NetMedic extends this idea of threshold by using the threshold to penalize a specific connection when the symptom value does not appear in its expected range as determined from the data.

NetMedic's Second Pass

After the Tallahassee meeting in March 1995, Robinson submitted a new knowledge table and two new sets of cause data. In addition, the signal processing engineers provided data for symptoms not reported before and updated symptom files based on newly modified signal processing routines.

Unable to run NetMedic directly with an expert on-site, the author utilized NetMedic in its default mode to determine the proposed connections. Susan Hruska then conducted a telephone interview with John Robinson, John Elling, Randy Roberts, and Sharbari Lahiri. Hruska conducted the conference call in the same manner as NetMedic operates: she discussed the current connection, proposed NetMedic's connection, prompted the expert for a decision, and recorded the result.

During the interview the experts could choose any connection type to add. If the original knowledge table had an ALWAYS and NetMedic proposed an INFREQUENTLY, during the conference call the experts could opt for a SOMETIMES. The results of NetMedic's second pass are contained in the tables that follow. Note the increased amount of symptoms reported.

Tabular Results of Interview

```
                        Table 8

        New NetMedic Findings for Column Degradation

                          NET-
Symptom               TABLE MEDIC FINAL ACTION
RetentionTimeShift      A     U     A   Confirm connection
SurrogatePrecision    None  None  None  Confirm
Spike Precision       None  None  None  Confirm
SensitivityChange     None    A   None  Augment Possible Signal Proc error
TailingPeaks            S     S     S   Confirm knowledge
UnresolvedPeaks         S     N     S   No data to find it
BandBroadening        None    I   None
ClippedPeaks          None    N   None
NegDipAfterPeak       None    N   None
IrregularBaseline     None    N   None
RisingBaseline          A     I     S   Refine Move toward proposal
CannotZeroBaseline    None  None  None  Confirm
HighNoise             None    I   None
HighBackground          S     N   None  Refine Move toward proposal
IrregularSpikes       None    N   None
GhostPeaks            None  None  None  Confirm
ExtraPeaks            None    N   None
NoPeaks               None    N   None
ReplicatePrecision    None    A   None  Augment Possible Signal Proc error
LeadingPeaks            S     S     S   Confirm knowledge
```

Table 8 illustrates further signal processing routine problems as seen in *Sensitivity Change*. *Unresolved Peaks* points to an incomplete data set; the data for *Column Degradation* had no symptom files with positive values for *Unresolved Peaks*, contrary to the experts' expectations. Table 8 shows again the level of disagreement between the data and the experts' knowledge. The entries for *Rising Baseline* show where NetMedic refined

knowledge, with the experts finally agreeing to a SOMETIMES connection. *High Background* illustrates NetMedic's capability to find connections which should be deleted.

Table 9

New NetMedic Findings for Leaking Septum

Symptom	TABLE	NET-MEDIC	FINAL	ACTION
RetentionTimeShift	A	A	A	Confirm connection
SurrogatePrecision	S	None	S	No data available
Spike Precision	S	None	S	No data available
SensitivityChange	A	A	A	Confirm knowledge
TailingPeaks	None	I	None	
UnresolvedPeaks	None	N	None	
BandBroadening	None	S	S	**Augment** New Knowledge
ClippedPeaks	None	N	None	
NegDipAfterPeak	None	N	None	
IrregularBaseline	S	N	S	
RisingBaseline	None	S	None	
CannotZeroBaseline	None	None	None	Confirm
HighNoise	None	S	S	**Augment** New Knowledge
HighBackground	None	N	None	
IrregularSpikes	None	N	None	
GhostPeaks	S	N	S	
ExtraPeaks	I	N	I	More data needed
NoPeaks	None	N	None	
ReplicatePrecision	S	A	U	**Refine** Move toward proposal
LeadingPeaks	None	S	None	

In Table 9 entries for *Band Broadening* indicate the discovery feature of NetMedic. Note also the number of NEVERs proposed but not accepted. *Replicate Precision* indicates a compromise in the refinement of the knowledge.

Table 10

New NetMedic Findings for Column Bleed

Symptom	TABLE	NET-MEDIC	FINAL	ACTION
RetentionTimeShift	I	S	I	
SurrogatePrecision	None	None	None	Confirm
Spike Precision	None	None	None	Confirm
SensitivityChange	None	A	None	
TailingPeaks	None	S	None	
UnresolvedPeaks	None	N	None	
BandBroadening	None	S	None	
ClippedPeaks	None	N	None	
NegDipAfterPeak	None	N	None	
IrregularBaseline	None	N	None	
RisingBaseline	A	A	A	Confirm knowledge
CannotZeroBaseline	None	None	None	Confirm

Table 10 -- continued

New NetMedic Findings for Column Bleed

Symptom	TABLE	NET-MEDIC	FINAL	ACTION
HighNoise	None	S	S	**Augment** New Knowledge
HighBackground	S	N	S	
IrregularSpikes	None	N	None	
GhostPeaks	None	None	None	Confirm
ExtraPeaks	None	N	None	
NoPeaks	None	N	None	
ReplicatePrecision	None	S	None	
LeadingPeaks	None	I	None	

Table 10 illustrates again new knowledge discovered by NetMedic with *High Noise*. The rejection by the experts of the NEVER connections proposed by NetMedic occurs often for this cause. For the causes *Leaking Syringe* and *Sample Too Concentrated* the experts considered several of NetMedic's suggestions but opted to stay with the original set of connections for these causes.

Table 11

New NetMedic Findings for Make-Up Gas Loss

Symptom	TABLE	NET-MEDIC	FINAL	ACTION
RetentionTimeShift	None	S	None	
SurrogatePrecision	None	None	None	Confirm
Spike Precision	None	None	None	Confirm
SensitivityChange	A	A	A	Confirm knowledge
TailingPeaks	S	S	S	Confirm knowledge
UnresolvedPeaks	None	N	None	
BandBroadening	S	S	S	Confirm knowledge
ClippedPeaks	S	N	S	
NegDipAfterPeak	None	N	None	
IrregularBaseline	None	I	None	
RisingBaseline	None	U	S	**Augment/refine**
CannotZeroBaseline	None	None	None	Confirm
HighNoise	None	S	S	**Augment** New Knowledge
HighBackground	S	N	I	**Refine** experts move near proposal
IrregularSpikes	None	N	None	
GhostPeaks	None	None	None	Confirm
ExtraPeaks	None	N	None	
NoPeaks	None	N	None	
ReplicatePrecision	None	A	None	
LeadingPeaks	None	I	None	

With the symptom *Rising Baseline* in Table 11, NetMedic discovered a new connection but the experts did not fully agree with the connection type proposed.

Performance Results

Performance of the new network showed significant improvement. The original network, represented in the tables above as **TABLE,** predicted 31 out of 107 causes correctly prior to training and 81 out of 107 after training. The latest table modified by NetMedic and accepted by the experts is represented by **FINAL** in the tables above. The latest knowledge table predicted 35 out of 107 causes correctly prior to training and 84 out of 107 correctly after training. NetMedic's network alone predicted 92 out of 107 correctly, even without training.

One might ask, "Why not use NetMedic's best network?" The danger lies in creating a network which is over-specific to a given data set. The current data is not all inclusive for GC fault data. NetMedic provides a fit to a given data set and if this data set is not representative of all possible faults and their data the network may not generalize well. The experts expressed confidence in the knowledge tables resulting from interaction between NetMedic and the experts. The experts believe that these tables better represent actual knowledge of the GC fault diagnosis process and that these tables will prove more effective with later data as the project continues.

Conclusion

Table 12 provides an overall summary of various knowledge tables/networks used in this research. The real data used in the majority of the tests as further differentiated into two groups: that originally produced by the signal processing team (Orig) and that produced using the improved signal processing routines (Imp).

Table 12

Overall Performance Summary

Date	Type data	#correct	#files	Description of table/network used
Dec 94	Ideal	76	76	Ideal simulated data generated to validate original CAEN network
Feb 95	Real-Orig	23	103	Original CAEN network developed from expert knowledge alone before NetMedic
Feb 95	Real-Orig	37	103	Original CAEN network with only four cause columns from the knowledge table implemented
Feb 95	Real-Orig	54	103	CAEN network w/ four causes and parallel paths removed
Mar 95	Real-Orig	74	103	NetMedic's automatic mode network without NEVERs proposed
Mar 95	Real-Orig	84	103	NetMedic's automatic mode network with NEVERs proposed
Mar 95	Real-Orig	94	103	NetMedic's automatic mode network with NEVERs, after training
Mar 95	Real-Orig	27	84	Experts table after using NetMedic with inappropriate data files removed
Mar 95	Real-Orig	40	84	Experts table after using NetMedic with NEVERs added and inappropriate data files removed
May 95	Real-Imp	31	107	Experts table described above without NEVERs on new data set from new signal processing routines
May 95	Real-Imp	81	107	Experts table described above after training
May 95	Real-Imp	35	107	Latest experts table after using NetMedic in second pass
May 95	Real-Imp	84	107	Latest experts table above after training
May 95	Real-Imp	92	107	NetMedic alone in second pass

Table 12 illustrates the progression of research and the resulting evolution of expert

knowledge for the problem domain. NetMedic significantly increased the CAEN network's

prediction capability during two major knowledge table revisions. NetMedic consistently

provided better networks and enabled the CAEN team to achieve some of their major project goals.

NetMedic demonstrated the ability to confirm, refine, and augment expert knowledge. NetMedic provides a data assisted method for knowledge acquisition that successfully alters table-based expert networks to increase prediction performance. The alteration maintains the ability to retrieve knowledge from the CAEN networks.

CHAPTER 8

FUTURE WORK

Among the many avenues for future exploration in the Contaminant Analysis Expert Network (CAEN) project using NetMedic, four main directions appear as the most promising. First, the method of determining what goes wrong is worthy of future research. When the network predicts incorrectly, what causes this? Second, how can the CAEN team better refine and make use of the threshold information? Third, how will NetMedic determine and handle severity of cause information? And lastly, how will NetMedic determine and analyze trend information?

Incorrect Network Predictions

How does one determine where, when, and why the network fails? Could the mistakes be caused by unknown parallel paths or by severely overlapping symptom sets?

Is a cause chosen incorrectly due to a conflict between symptoms for this cause indicating an undiscovered parallel path? One could detect these parallel paths easily by looking for groupings in the data with NetMedic. These groupings would be widely dissimilar in the symptom values when the network predicts correctly and those values where it predicts incorrectly. One could analyze these groupings of values to determine if these parallel paths exist.

66

Is the network predicting incorrectly due to extremely similar symptom sets leading to two different causes? When the network predicts incorrectly, does the network consistently predict a particular cause? If so, do the predicted and expected cause share a similar symptom set? How can the CAEN team build their network to discriminate between causes with overlapping symptom sets? Future extensions to NetMedic could help determine the answers to these questions.

Thresholds

Currently NetMedic alters thresholds on various connections to tighten the window of acceptable values for a given symptom and a given cause. To improve network performance by changing the functionality of the threshold, researchers could either work within the framework of CAEN networks or change the inferencing in these networks.

Current Use of Thresholds

The thresholds in CAEN networks currently act as floors, above which all symptom values contribute in a positive manner. See Figure 9 on the following page.

As illustrated, these thresholds help narrow somewhat the range of values expected. The figure illustrates how a single symptom's values relate to each of the six different causes. The current use of thresholds aids the network in choosing cause B over Cause A if the symptom value was 0.25. Note that the complement is not true: if the symptom value was 0.75, the network would not have any additional information to select cause A over cause B since 0.75 is above the threshold for both causes.
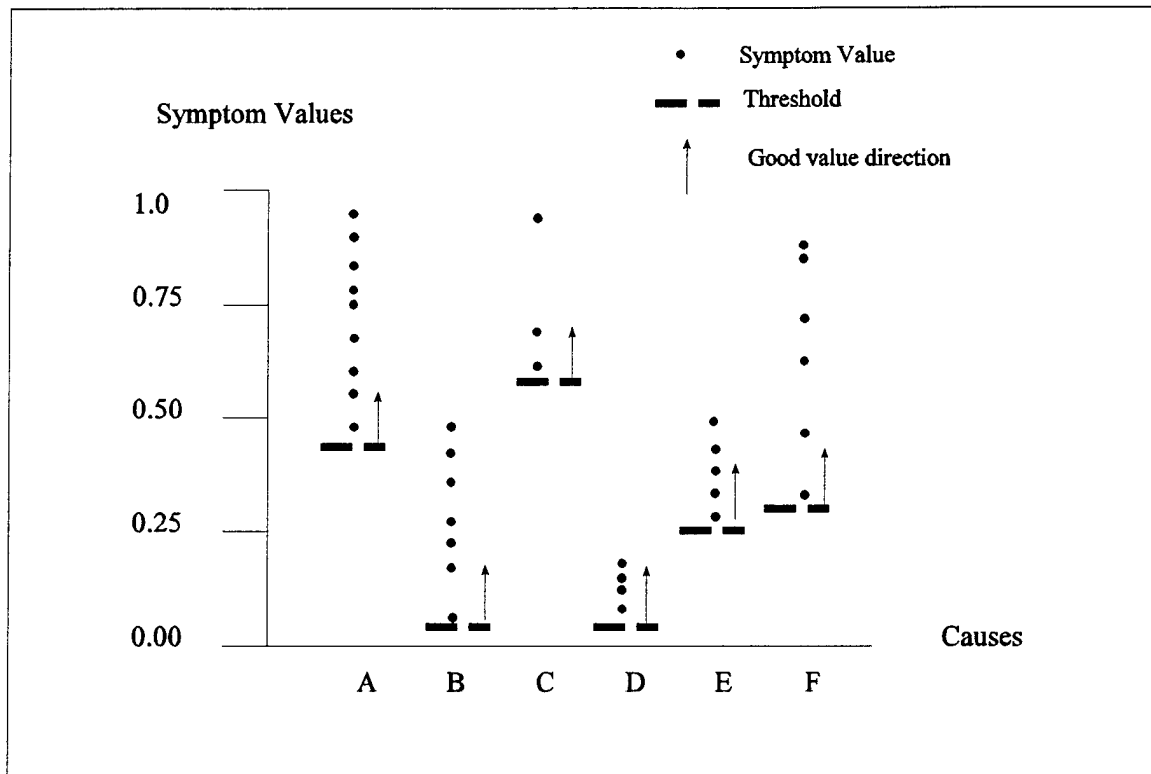
Figure 9: **Current use of Thresholds**.

Changing the Functionality of Thresholds in Current CAEN Networks

One idea involves using the threshold to partition the causes such as those illustrated in Figure 9 into two subsets based on their related symptom values. With six causes worth of data available, NetMedic could be modified to partition the causes into a high group and a low group based on the related symptom's mean and standard deviation as illustrated in Figure 10.

After modification, NetMedic could adjust the thresholds for the high group normally. NetMedic could then adjust the threshold for the low group to the mean plus one standard deviation, imposing a ceiling for the lower group. NetMedic could propose exchanging the lower group's Good-Dog and Bad-Dog factors and thus partition the sets by the changed

functionality of their connections in the CAEN networks. The reversed Good-Dog and Bad-Dog values result in the lower group causes receiving positive values as long as the related symptom value falls below the threshold.
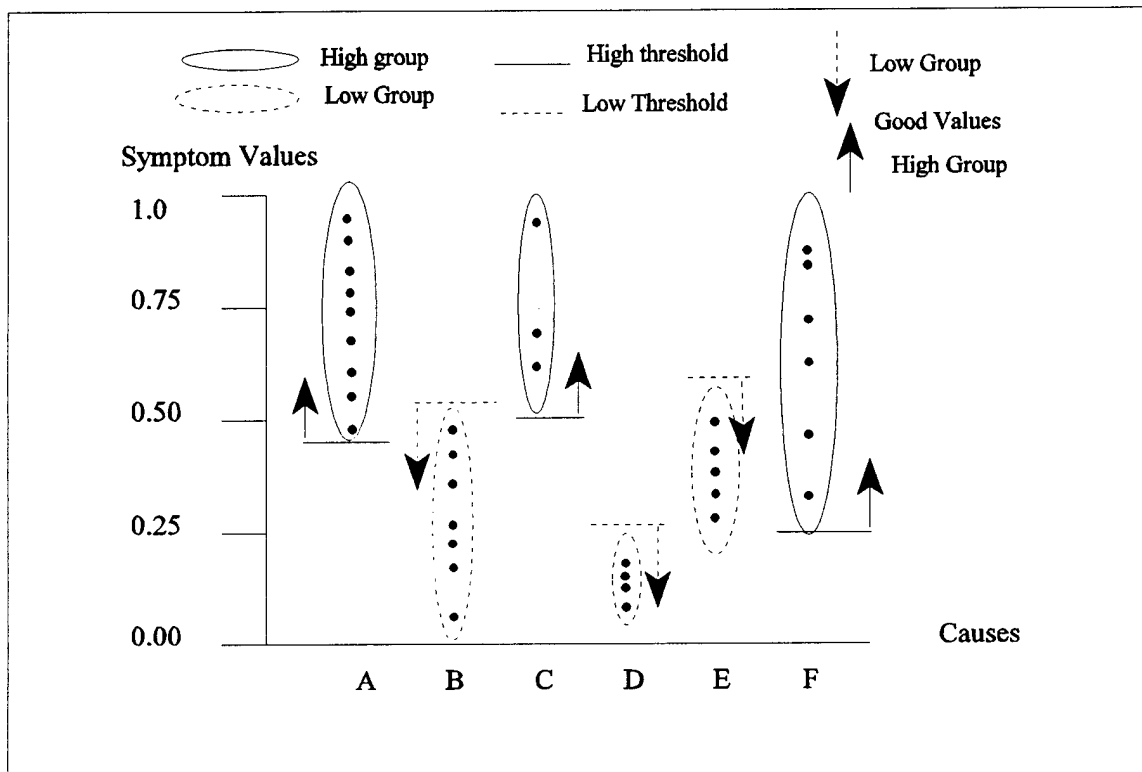
Figure 10: One Possible Extended use of Thresholds.

This modification of the use of thresholds could still result in overlapping symptom ranges within the high and low groups. The modification does further refine the prediction capability by providing more means of discrimination for the network without altering the inferencing mechanism.

Changing Threshold Functionality with New Inferencing

Another idea is to implement an upper and lower threshold for the symptom for each cause. With upper and lower thresholds, a symptom would have a range or *window* of values

for a given cause. If the input symptom value fell within this range for a cause, this would contribute toward prediction of that cause in a positive manner. Otherwise, a negative contribution for that symptom towards concluding the fault would result.

An additional method would involve altering the CAEN inferencing by adding two Filter nodes for each symptom related to a cause (Figure 11). NetMedic would adjust one Filter node as normal, and adjust the other as described for the lower group above, creating Filter nodes with opposite functionality. A new Filter Combination layer would be added that interprets the outputs of these paired Filter nodes. If both new Filter nodes have positive output, the Filter Combination layer would fire positively. If both new Filter nodes do not

Propagates a positive value if
Symptom value exceeds threshold

Normal Filter Node

Symptom Node

New Filter Combination Node

Propagates a positive
value if both inputs
are positive

Opposite Filter Node
Propagates a positive value if
Symptom value is below threshold
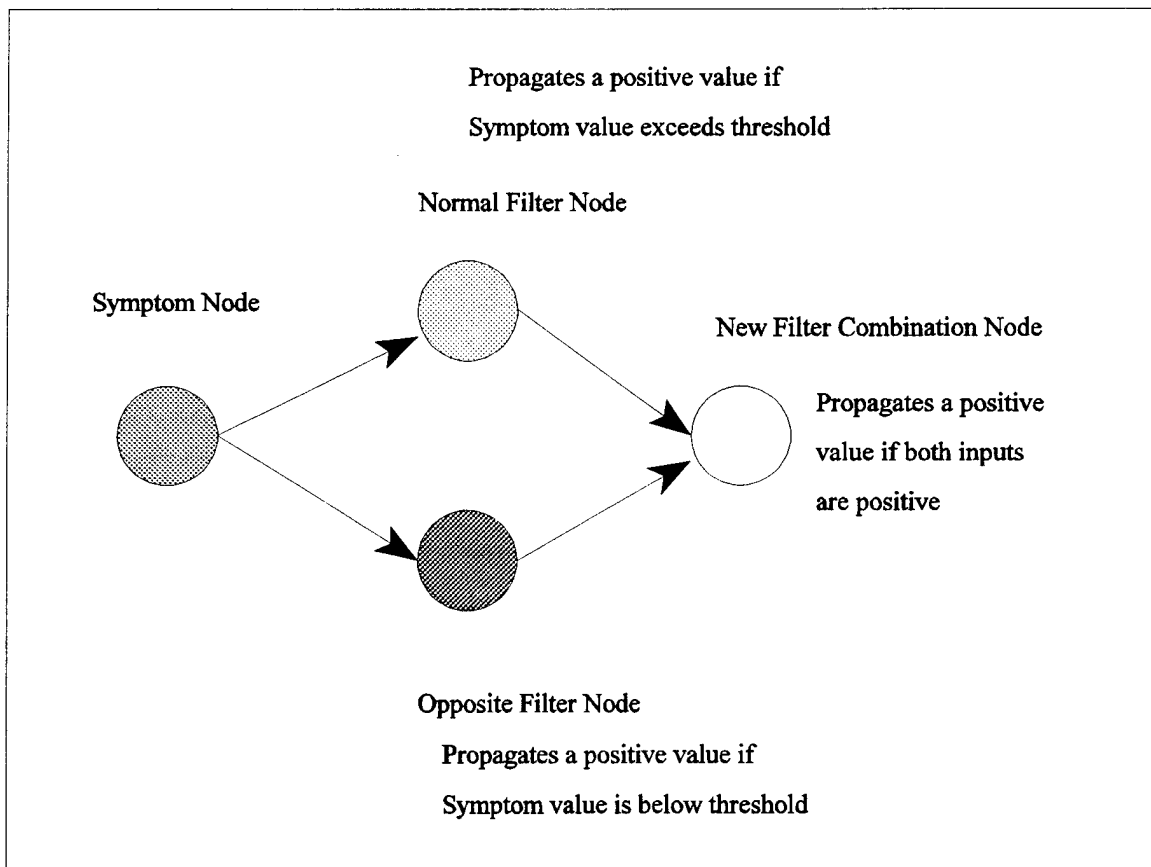
Figure 11: Modified CAEN Network.

have positive output, the Filter Combination layer could propagate some computed percentage value.

The new Filter Combination nodes would effectively provide a threshold and ceiling for each Symptom node and could aid in separating the data space. See Figure 12. The new Filter Combination node would connect to the current Combination nodes.
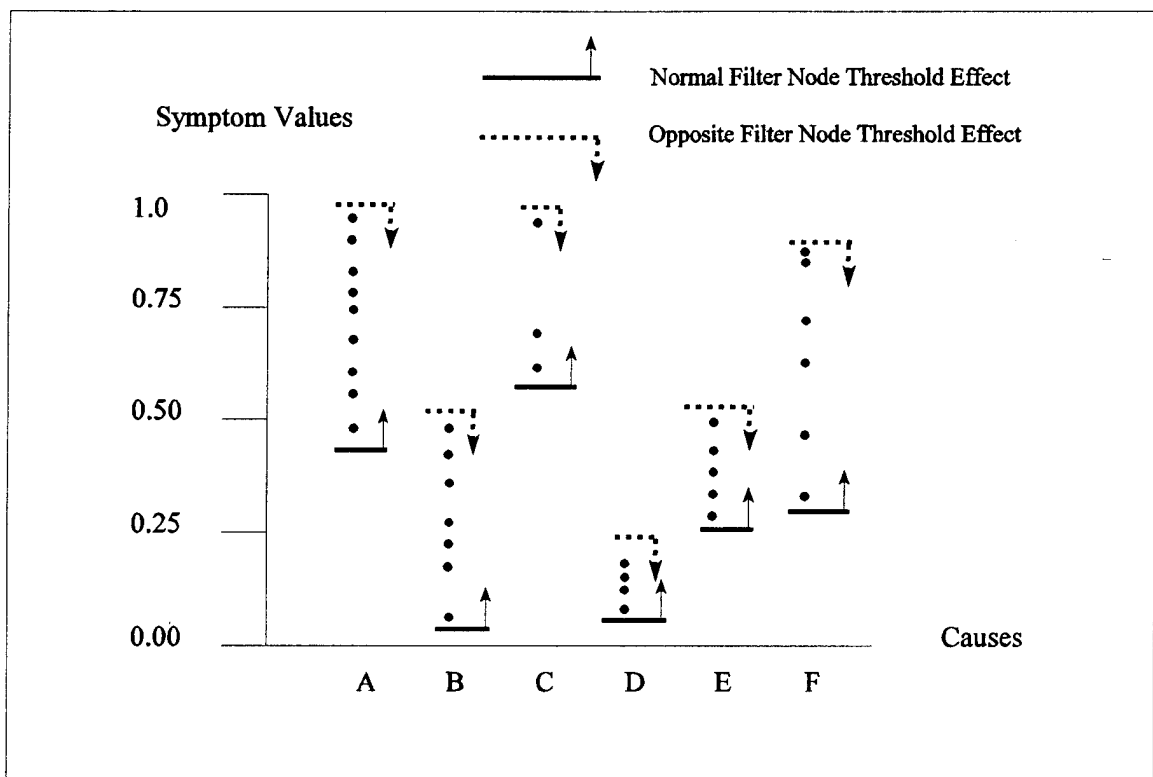
Figure 12: Another look at Thresholds.

The altered inference mechanism would give additional discriminatory power to the network. Overlap of symptom value ranges will still occur, but the modification described would provide better discrimination ability in the network.

Cause Severity Information

The review of correlation of cause severity information in relation to symptom values is another promising avenue for further research. If this correlation exists, what is implied about the connection types? Would this merely aid in the training algorithm? Currently, the CAEN networks nominate the cause with the highest value. Interpreting this value as it relates to a measure of severity would prove valuable as feedback to the experts and as additional information for the operators using the automated fault diagnosis system.

Trend Information

Finally, some causes are affected by the length of time the process has been running. For example, if there have been 200 samples processed since the machine had the septum changed, *Leaking Septum* becomes very likely. NetMedic could capture and use this information. Currently the CAEN networks and NetMedic implement no real style of trend analysis. The signal processing team deals with this trend information somewhat in their symptom detection routines.

Conclusion

Future modifications to NetMedic proposed would allow the CAEN team to determine the reasons for poor predictions, utilize thresholds differently for better performance, incorporate cause severity information into the network, and exploit historical trend information. The potential for increased performance using extensions of the NetMedic tool is great.

CHAPTER 9

CONCLUSIONS

NetMedic provides a tool to experts for *confirming*, *refining*, and *augmenting* their knowledge. NetMedic has proven to be effective in capturing fault diagnosis knowledge in gas chromatography by suggesting several new relationships between symptoms and causes of failures in these instruments. The Contaminant Analysis Expert Network (CAEN) team gained major insights from the knowledge acquisition process using NetMedic as a dialogue tool between the knowledge engineers and the human experts. The process of modifying expert networks on the fly based on statistical data searches is also new.

Tool for Knowledge Acquisition

NetMedic aids in the knowledge acquisition process by confirming existing connections based on the relationships found in data. It refines expert knowledge by altering the functionality of existing relationships. It augments expert knowledge by discovering new relationships and possible problems with signal processing routines creating the symptom files from the gas chromatograms.

NetMedic functions using a set of symptom data files with fault diagnoses. NetMedic uses values determined by a CAEN network. NetMedic processes this set of data into a set of descriptive statistics which is used to propose connections between symptoms and causes.

NetMedic can operate in one of two modes: interactive or automatic. The interactive mode prompts the expert to make decisions about proposed connections in the networks that relate directly to entries in the knowledge tables. In the automatic mode, NetMedic determines the connections and builds them into the network.

Results

NetMedic demonstrated the ability to confirm, refine, and augment expert knowledge in the area of fault diagnosis of GC data during the CAEN team's ongoing research. In addition to producing CAEN networks with increased performance, NetMedic preserves the captured expert knowledge and reasoning when altering the network architecture. Knowledge engineers used the new expert knowledge NetMedic discovered on several occasions, improving the knowledge they were able to extract from the expert alone. NetMedic also provided highly effective feedback to the signal processing engineers on the accuracy of their algorithms. NetMedic is a useful and functional tool and has proven valuable to the ongoing Contaminant Analysis Automation project. The principles inherent in NetMedic's use of data-assisted methods to improve systems built to capture expert knowledge are far-reaching.

APPENDIX A

MAY 1995 KNOWLEDGE TABLE

This appendix contains the latest knowledge table in use by the Contaminant Analysis Expert Network (CAEN) team. Input to this knowledge table was provided by the CAEN members, John Robinson of Varian Equipment Corporation, John Elling, Sharbari Lahiri, and Randy Roberts.

The knowledge table utilizes the AUSIN entries described in this thesis. The table contains rows relating to the symptoms that may appear. The columns represent causes relating to faults in Gas Chromatography (GC) instruments. The experts further grouped the causes into one of six main areas: Carrier Gas, Injection, Injector, Column, Fuel Gas, and Detector. These six areas relate to major component areas of the Gas Chromatography equipment.

The row and column intersections form cells which may or may not have an entry. A blank entry indicates no connection between symptom and cause while an "A" indicates that the symptom will always appear in a chromatogram if the given cause is present. A "U" indicates a USUALLY relationship; an "S" indicates a SOMETIMES relationship; and an "I" indicates an INFREQUENTLY relationship.

For example, for symptom *Spike Precision* and cause *Leaking Syringe* there is a "U" entry. This indicates the expert's opinion that if *Leaking Syringe* occurs, the symptom *Spike*

75

Precision will usually appear in the chromatogram.

An "N" entry indicates a definite negative relationship between a given symptom and cause. The experts developed these entries by taking a cause and determining which symptoms would definitely not appear in a chromatogram. In the latest knowledge table there are no "N" entries.

The shaded cause columns indicate the causes for which the CAEN team has real GC symptom data. The signal processing engineers have algorithms to extract from chromatograms the symptoms indicated by shaded rows.

CAA/DIM symptom/cause relationships
Adapted from DIM expert information and Varian, Diagnosis/Troubleshooting Guide

Dated May 1995

Goals:

Causes (by group) / **Symptoms**

Symptoms	Injector: 1. Leaking Syringe	Injector: 2. Dirt in the Syringe	Injector: 3. Sample Decomposition	Column: 4. Late Elution	Column: 5. Column Bleed	Column: 6. Column Degradation	Column: 7. Contaminated Column	Detector: 8. Contaminated Detector	Detector: 9. Wrong Fuel Gas Flow (ECD)	Detector: 10. Dust in Flame (FID)	Detector: 11. Make Up Gas Loss (ECD)	Injection: 12. Dirt in the Injector	Injection: 13. Leaking Septum	Sample: 14. Sample Too Concentrated	Sample: 15. Sample Size Too Large	Sample: 16. Sulphur Contamination	Sample: 17. Phalate Contamination	Fuel Gas: 18. Carrier Gas Low
1. Retention Time Shift	U	U		U		A	A					I	A		S			A
2. Spike Precision	U												S					
3. Surrogate Precision	A		S		I								S					
4. Sensitivity Change					A	A		A	A		A	S	A					U
5. Tailing Peaks					S	S					S	S		S	S			S
6. Unresolved Peaks				A		S					S	S		S	A			S
7. Band Broadening														A	A			
8. Clipped Peaks				A										A	A			
9. Negative Dip After Peak								U										
10. Irregular Baseline					A	S	S	U			S		S					
11. Rising Baseline					S	S	S	U			S	S	S					
12. Cannot Zero Baseline							U											
13. High Noise								S			S	S		S	S			S
14. High Background	A		U				S	U			S	S	S	A				
15. Irregular Spikes			U	A	S		S			A	I	S	S		S	A	A	
16. Ghost Peaks		S											S					
17. Extra Peaks	U	A	U	A			S					S	S	S	S			
18. No Peaks	I		U										I					I
19. Replicate Precision	U	U	U										S					
20. Leading Peaks	U	S	S			S						U	U		S			I

APPENDIX B

CAEN SYMPTOM INPUT FILE

The following page contains the standard .gfi (Gensym File Interface) file used by the Contaminant Analysis Expert Network (CAEN) networks as an input file. Comment lines in the file begin with a ';'. The file contains a time stamp that G2 utilizes to order specific files. The file consists of lines with the following format: a zero followed by a tab, followed by a field name, followed by a tab, followed by a field value. The CAEN G2 application recognizes the field names and tries to match them to corresponding input variable names. If a match occurs, the input variable receives the value from the file.

The file has various information describing the type of sample, source of the file, the induced or expected cause, and the symptom values for a gas chromatogram. The signal processing team uses their routines to extract the symptoms from gas chromatograms in an automated method and produces this file.

```
;SYMPTOM FILE FOR DEGR055.cdf
13 Apr 1995 17:31:41        the current time    0
;SAMPLE INFO
 0          sample_name      n/a
 0          sample_id        0
 0          sample_type      Calibration
;FILE INFO
 0          File_source          GCProcCalibration v1.1
 0          Reference_gc_filename     DEGR052.CDF
 0          Retention_time_filename   DEGR_C.RTM
;INDUCED CAUSE
 0          Induced_cause     ColumnDegradation
 0          Severity 0.0
;SYMPTOMS
 0          ClippedPeaks        0.000
 0          NoPeaks             0.000
 0          RisingBaseline      0.000
 0          IrregularBaseline 0.000
 0          CannotZeroBaseline          -2.000
 0          TailingPeaks        0.000
 0          LeadingPeaks        0.360
 0          UnresolvedPeaks 0.000
 0          GhostPeaks          -2.000
 0          ExtraPeaks          0.000
 0          NegativeDipAfterPeak        0.000
 0          IrregularSpikes     0.000
 0          SensitivityChange           0.960
 0          RetentionTimeShift          0.750
 0          BandBroadening      0.000
 0          SpikePrecision      -2.000
 0          SurrogatePrecision          -2.000
 0          ReplicatePrecision          0.920
 0          HighNoise           0.000
 0          HighBackground 0.000
```

APPENDIX C

NETMEDIC INPUT FILE

This appendix contains the input file NetMedic uses. The author developed the NetMedic input file to standardize NetMedic's input. The NetMedic input file contains a subset of the `.gfi` file's information along with some new information.

The NetMedic input file begins with the induced/expected cause preceded by "EXP." The cause actually predicted the highest by the CAEN network follows, preceded by "PRED."

The file then breaks down into a cause section and a symptom section. The cause section begins with the text string "CAUSES" followed by the number of causes contained in the network. Each line that follows lists a cause index, the name of the cause and the value computed by the network for that cause. The cause section ends with a lone -1.

The symptom section begins with the text string "SYMPTOMS" followed by the number of symptoms found in the network. Each line that follows contains a symptom's index, the symptom name, and the value for that symptom. The symptom section concludes with an -1 on a separate line.

These files relate one to one to the files described in Appendix B. Future work will modify the `.gfi` files to contain the information needed by NetMedic. The modification

involves appending the `.gfi` files with the cause expected, the cause actually predicted, and

the cause values.

```
EXP ColumnDegradation
PRED LEAKINGSYRINGE
CAUSES
7
0   LEAKINGSYRINGE 0.943
1   COLUMNBLEED -0.24
2   COLUMNDEGRADATION 0.269
3   MAKEUPGASLOSS 0.796
4   LEAKINGSEPTUM 0.799
5   SAMPLETOOCONCENTRATED -0.431
6   ERASEME -0.015
-1
SYMPTOMS
20
0   RETENTIONTIMESHIFT 0.75
1   SPIKEPRECISION -2.0
2   SURROGATEPRECISION -2.0
3   SENSITIVITYCHANGE 0.96
4   TAILINGPEAKS 0.0
5   UNRESOLVEDPEAKS 0.0
6   BANDBROADENING 0.0
7   CLIPPEDPEAKS 0.0
8   IRREGULARBASELINE 0.0
9   RISINGBASELINE 0.0
10   HIGHBACKGROUND 0.0
11   GHOSTPEAKS -2.0
12   EXTRAPEAKS 0.0
13   NOPEAKS 0.0
14   REPLICATEPRECISION 0.92
15   LEADINGPEAKS 0.36
16   CANNOTZEROBASELINE -2.0
17   NEGATIVEDIPAFTERPEAK 0.0
18   IRREGULARSPIKES 0.0
19   HIGHNOISE 0.0
-1
```

APPENDIX D

CAEN NETWORK PICTURE

The following page contains a screen dump of a portion of the Contaminant Analysis Expert Network (CAEN) networks implemented in Gensym's G2. The network processes information from left to right.

The symbols on the left represent Symptom or input nodes. The second layer of symbols represent our Filter nodes with the letter in the center of the icon identifying the type of Filter node. The next layer consists of Combination nodes whose icons have the radiation safety symbol. The Combination nodes combine the evidence associated with a particular cause. The Cause node receives input from the Combination nodes and computes the final value reported by the network.

The blue arrows correspond to a symptom having a zero value and the zero value being propagated along that connection. The green arrows indicate positive values which the network propagates from one layer to the next.
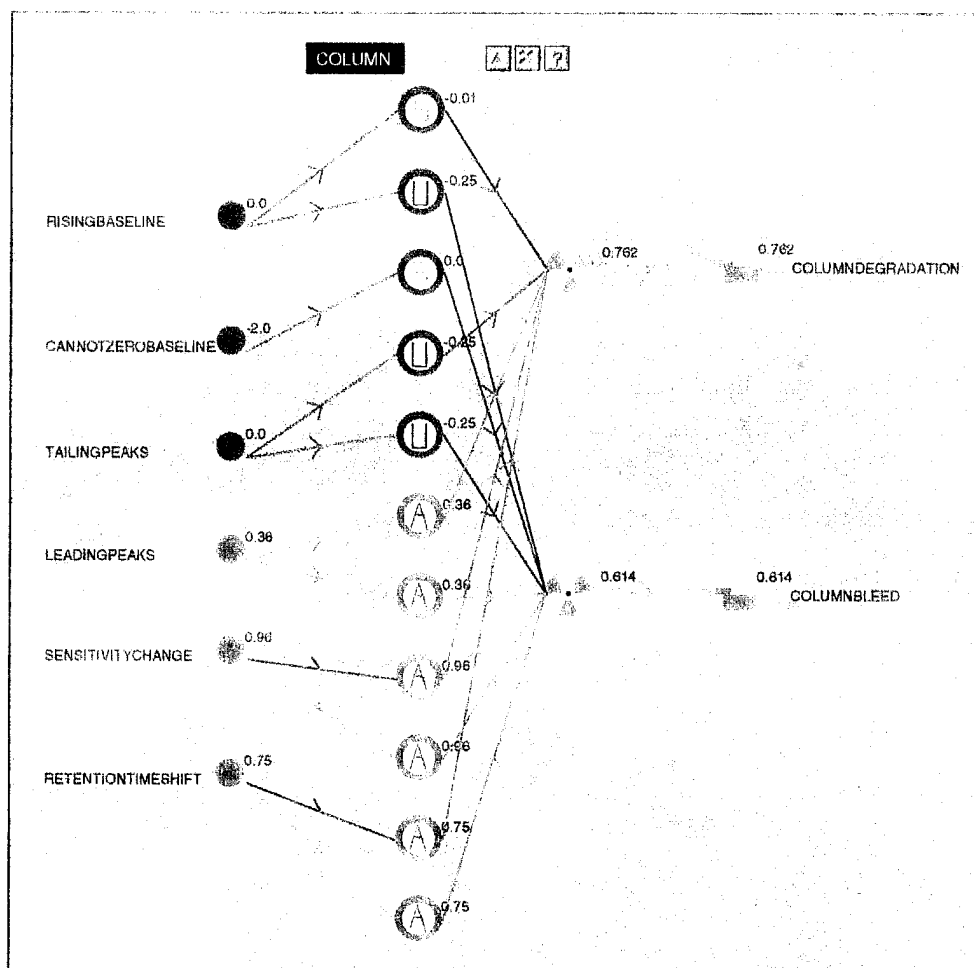
APPENDIX E

COMMON NETWORK FILE

This appendix contains the latest common network file representation of the latest knowledge table (Appendix B). The file consists of three sections.

The first section is for comments. All lines in the first section begin with a '#' to indicate a comment line.

The next section represents the connections from Symptom nodes to Combination nodes. The format for each entry is broken down into eight fields. The format represents the four layers of the network in just two layers of connections: from Symptom through Filter to Combination and from Combination to Cause.

In the Symptom to Combination section of the file, the first field in each line is a text string containing the name of the workspace on which this connection appears. For example, the text string relates a given symptom connection to a specific cause workspace. The first symptom entry in the file has "COLUMN" as its first field. The first field identifies this connection as belonging to a cause which falls in the Column subgroup.

The second field on each line is a text string containing the name of the symptom. The third field is a split alphabetic and numeric field. The first character is a 'C', identifying it as a Combination node. The second subfield is the number of the designated Combination node. Due to the presence of multiple paths to a given Cause in the CAEN network, the numbering

84

scheme is used to identify multiple paths. The fourth field is a floating point number that represents the weight on the connection from the Symptom node to the Filter node. Currently these are all 1.000000. All the weight values fall within the -1.000000 to 1.000000 range.

The fifth field in a line from this section represents the threshold for the Filter node in this connection. The sixth field is the Good-Dog factor and falls in the range -1.000000 to 1.000000. The seventh field represents the Bad-Dog factor and falls between -1.000000 and 1.000000. The eighth and final field is the connection weight between the Filter node and the Combination node in the next layer. One can identify these Symptom to Combination node entries by the second field being a symptom name.

The third section of the file contains information for Combination node to Cause node connections. One can differentiate the third section from the second section by the second field in each line. The third section has a different format than the Symptom to Combination node section. There are four fields in an entry in the third section. The first field is the workspace name, the same as for the Symptom to Combination node section.

The second field indicates the Combination node related to this cause. The first part is a character, the letter 'C'. If parallel paths to a particular cause exist, there will be two file entries in this section with identical Cause names but different Combination node numbers. The third field is a text string containing the Cause name. The fourth field contains a floating point number from -1.000000 to 1.000000 representing the connection weight between the Combination node and the Cause node.

johnbobtable.cn - Common.net format created from johnbobtable.tab by table2common.
COLUMN RETENTIONTIMESHIFT C1 1.000000 0.000000 1.000000 -0.001000 0.250000
COLUMN RETENTIONTIMESHIFT C2 1.000000 0.000000 1.000000 -0.500000 0.850000
INJECTION RETENTIONTIMESHIFT C4 1.000000 0.000000 1.000000 -0.500000 0.850000
INJECTOR SPIKEPRECISION C0 1.000000 0.000000 1.000000 -0.250000 0.750000
INJECTION SPIKEPRECISION C4 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTOR SURROGATEPRECISION C0 1.000000 0.000000 1.000000 -0.250000 0.750000
INJECTION SURROGATEPRECISION C4 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTOR SENSITIVITYCHANGE C0 1.000000 0.000000 1.000000 -0.500000 0.850000
INJECTION SENSITIVITYCHANGE C3 1.000000 0.000000 1.000000 -0.500000 0.850000
INJECTION SENSITIVITYCHANGE C4 1.000000 0.000000 1.000000 -0.500000 0.850000
COLUMN TAILINGPEAKS C2 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION TAILINGPEAKS C3 1.000000 0.000000 1.000000 -0.010000 0.500000
COLUMN UNRESOLVEDPEAKS C2 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION UNRESOLVEDPEAKS C5 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION BANDBROADENING C3 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION BANDBROADENING C5 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION CLIPPEDPEAKS C3 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION CLIPPEDPEAKS C5 1.000000 0.000000 1.000000 -0.500000 0.850000
INJECTION IRREGULARBASELINE C4 1.000000 0.000000 1.000000 -0.010000 0.500000
COLUMN RISINGBASELINE C1 1.000000 0.000000 1.000000 -0.500000 0.850000
COLUMN RISINGBASELINE C2 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION RISINGBASELINE C3 1.000000 0.000000 1.000000 -0.010000 0.500000
COLUMN HIGHNOISE C1 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION HIGHNOISE C3 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION HIGHNOISE C4 1.000000 0.000000 1.000000 -0.010000 0.500000
COLUMN HIGHBACKGROUND C1 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION HIGHBACKGROUND C3 1.000000 0.000000 1.000000 -0.001000 0.250000
INJECTOR GHOSTPEAKS C0 1.000000 0.000000 1.000000 -0.250000 0.750000
INJECTION GHOSTPEAKS C4 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION GHOSTPEAKS C5 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTION EXTRAPEAKS C4 1.000000 0.000000 1.000000 -0.001000 0.250000
INJECTOR NOPEAKS C0 1.000000 0.000000 1.000000 -0.001000 0.250000
INJECTOR REPLICATEPRECISION C0 1.000000 0.000000 1.000000 -0.250000 0.750000
INJECTION REPLICATEPRECISION C4 1.000000 0.000000 1.000000 -0.250000 0.750000
COLUMN LEADINGPEAKS C2 1.000000 0.000000 1.000000 -0.010000 0.500000
INJECTOR C0 LEAKINGSYRINGE 1.000000
COLUMN C1 COLUMNBLEED 1.000000
COLUMN C2 COLUMNDEGRADATION 1.000000
INJECTION C3 MAKEUPGASLOSS 1.000000
INJECTION C4 LEAKINGSEPTUM 1.000000
INJECTION C5 SAMPLETOOCONCENTRATED 1.000000

APPENDIX F

NETMEDIC STATISTICS FILE

This appendix contains a portion of the descriptive statistics file generated by NetMedic. This portion reports the information for only one cause. The entire statistics file would contain information for all causes represented in the set of data files. A cause section begins with the cause name. Immediately below the cause name are the overall statistics for that particular cause.

The information for a cause consists of the total number of files where the induced fault is this cause, the total number of times the network predicted the cause correctly and the number of times the network predicted incorrectly. In addition, the file lists the mean of the cause values computed by the network for three distinct groups: the mean for all the files for this cause, the mean of the cause value when the network predicts this cause correctly, and the mean when the network predicts incorrectly. The file displays all information in these three main areas labeled as total, correct, and incorrect. The cause section also contains the standard deviation for the cause values. NetMedic currently reports these cause means for use by the expert in evaluating NetMedic's proposals.

The next part of a cause section represents the values of a symptom when this cause occurs. This part of the file begins with the symptom name and reports results for the total section, good section, and bad section as described above. The file contains the mean of

symptom values, the standard deviation of these values, the total number of times the symptom was positive (here non-zero), the number of times the symptom had a zero value (did not appear), the number of times the symptom value was a negative one and the number of times the symptom was a negative two.

NetMedic utilizes the statistical information to determine connections from the data (see Chapter 6).

```
∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧
= NetMedic STATISTICS FILE
= This file contains the statistics derived from :
=       GENERATION    DATE         Tue   Jun    6   08:08:35   1995=    INPUT   FILE   NAME
/home/cs35/timpany/CAENDATA/temp/bob-output
∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧∧
```

Cause LEAKINGSYRINGE

Total number 25	Total good 23	Total bad 2
Total mean 0.764640	Mean Good : 0.763565	Mean Bad: 0.777000
Total dev 0.363648	Dev Good : 0.378212	Dev Bad: 0.162635

RETENTIONTIMESHIFT

Total Mean: 0.330	Good Mean: 0.280	Bad Mean: 0.905
Total Dev: 0.311	Good Dev: 0.269	Bad Dev: 0.134
Total Corr: 0.479	Good Corr: 0.565	Bad Corr: 1.211
Total Num Pos: 17	Good Num Pos: 15	Bad Num Pos: 2
Total Num Zero: 8	Good Num Zero: 8	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

SPIKEPRECISION

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 0	Good Num Zero: 0	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 25	Good Num -2: 23	Bad Num -2: 2

SURROGATEPRECISION

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 0	Good Num Zero: 0	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 25	Good Num -2: 23	Bad Num -2: 2

SENSITIVITYCHANGE

Total Mean: 0.775	Good Mean: 0.770	Bad Mean: 0.835
Total Dev: 0.403	Good Dev: 0.418	Bad Dev: 0.233
Total Corr: 0.973	Good Corr: 0.996	Bad Corr: -0.697
Total Num Pos: 20	Good Num Pos: 18	Bad Num Pos: 2
Total Num Zero: 0	Good Num Zero: 0	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 5	Good Num -2: 5	Bad Num -2: 0

TAILINGPEAKS

Total Mean: 0.323	Good Mean: 0.336	Bad Mean: 0.170
Total Dev: 0.369	Good Dev: 0.379	Bad Dev: 0.240
Total Corr: 0.475	Good Corr: 0.474	Bad Corr: 0.676
Total Num Pos: 13	Good Num Pos: 12	Bad Num Pos: 1
Total Num Zero: 12	Good Num Zero: 11	Bad Num Zero: 1
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

UNRESOLVEDPEAKS

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 25	Good Num Zero: 23	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

BANDBROADENING

Total Mean: 0.046	Good Mean: 0.050	Bad Mean: 0.000
Total Dev: 0.161	Good Dev: 0.167	Bad Dev: 0.000
Total Corr: 0.163	Good Corr: 0.166	Bad Corr: -2.000
Total Num Pos: 2	Good Num Pos: 2	Bad Num Pos: 0
Total Num Zero: 23	Good Num Zero: 21	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

CLIPPEDPEAKS

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 25	Good Num Zero: 23	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

IRREGULARBASELINE

Total Mean: 0.800	Good Mean: 0.783	Bad Mean: 1.000
Total Dev: 0.408	Good Dev: 0.422	Bad Dev: 0.000
Total Corr: 0.986	Good Corr: 1.000	Bad Corr: -2.000
Total Num Pos: 20	Good Num Pos: 18	Bad Num Pos: 2
Total Num Zero: 5	Good Num Zero: 5	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

RISINGBASELINE

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 25	Good Num Zero: 23	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

HIGHBACKGROUND

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 20	Good Num Zero: 18	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 5	Good Num -2: 5	Bad Num -2: 0

GHOSTPEAKS

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 0	Good Num Zero: 0	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 25	Good Num -2: 23	Bad Num -2: 2

EXTRAPEAKS

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 25	Good Num Zero: 23	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

NOPEAKS

Total Mean: 0.200	Good Mean: 0.217	Bad Mean: 0.000
Total Dev: 0.408	Good Dev: 0.422	Bad Dev: 0.000
Total Corr: -0.986	Good Corr: -1.000	Bad Corr: -2.000
Total Num Pos: 5	Good Num Pos: 5	Bad Num Pos: 0
Total Num Zero: 20	Good Num Zero: 18	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

REPLICATEPRECISION

Total Mean: 0.760	Good Mean: 0.783	Bad Mean: 0.500
Total Dev: 0.436	Good Dev: 0.422	Bad Dev: 0.707
Total Corr: 0.950	Good Corr: 1.000	Bad Corr: 0.230
Total Num Pos: 19	Good Num Pos: 18	Bad Num Pos: 1
Total Num Zero: 6	Good Num Zero: 5	Bad Num Zero: 1
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

LEADINGPEAKS

Total Mean: 0.039	Good Mean: 0.043	Bad Mean: 0.000
Total Dev: 0.064	Good Dev: 0.066	Bad Dev: 0.000
Total Corr: 0.333	Good Corr: 0.342	Bad Corr: -2.000
Total Num Pos: 7	Good Num Pos: 7	Bad Num Pos: 0
Total Num Zero: 18	Good Num Zero: 16	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

CANNOTZEROBASELINE

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 0	Good Num Zero: 0	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 25	Good Num -2: 23	Bad Num -2: 2

NEGATIVEDIPAFTERPEAK

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 25	Good Num Zero: 23	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

IRREGULARSPIKES

Total Mean: 0.000	Good Mean: 0.000	Bad Mean: 0.000
Total Dev: 0.000	Good Dev: 0.000	Bad Dev: 0.000
Total Corr: -2.000	Good Corr: -2.000	Bad Corr: -2.000
Total Num Pos: 0	Good Num Pos: 0	Bad Num Pos: 0
Total Num Zero: 25	Good Num Zero: 23	Bad Num Zero: 2
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 0	Good Num -2: 0	Bad Num -2: 0

HIGHNOISE

Total Mean: 0.736	Good Mean: 0.713	Bad Mean: 1.000
Total Dev: 0.426	Good Dev: 0.437	Bad Dev: 0.000
Total Corr: 0.861	Good Corr: 0.878	Bad Corr: -2.000
Total Num Pos: 20	Good Num Pos: 18	Bad Num Pos: 2
Total Num Zero: 0	Good Num Zero: 0	Bad Num Zero: 0
Total Num -1: 0	Good Num -1: 0	Bad Num -1: 0
Total Num -2: 5	Good Num -2: 5	Bad Num -2: 0

APPENDIX G

REPORT OF NETMEDIC CHANGES

TO COMMON.NET FILE

The following pages contain a portion of the file that NetMedic uses to record alterations to the network file during interaction with the expert. The file consists of sections for each cause modified by an expert. Within each cause section, there are two additional subsections. The first subsection relates to NetMedic's comparison of connections already contained in the `common.net` file. The second subsection consists of those connections proposed by NetMedic that were not in the current `common.net` file.

In the first section, the file contains the differences between the current and proposed connection. The reasoning for the proposed connection follows with the resultant change/modification chosen by the expert. The second subsection contains the results of the expert's decisions on proposed connections which do not exist in the `common.net` being modified.

```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
=  NetMedic RECORDING FILE
=  This file contains the decisions made by :
=  EXPERT              Robert G. Timpany
=  MODIFICATION DATE   Tue Jun  6 08:08:35 1995
=  STATISTICS FILE NAME /home/cs35/timpany/CAENDATA/temp/results.out
=  ORIGINAL COMMON NET FILE /home/cs35/timpany/CAENDATA/temp/johnstable.cn
=  MODIFIED COMMON NET FILE /home/cs35/timpany/CAENDATA/temp/newtable.cn
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
==================================================
CAUSE SECTION FOR CAUSE COLUMNDEGRADATION
==================================================
```

```
- - - - - - - - - - - - - - - - - - - -
SYMPTOM  RETENTIONTIMESHIFT
- - - - - - - - - - - - - - - - - - - -
```

●Current Connection for Symptom RETENTIONTIMESHIFT
to Cause COLUMNDEGRADATION is an ALWAYS

●Proposed Connection for Symptom RETENTIONTIMESHIFT
to Cause COLUMNDEGRADATION is an USUALLY

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 3.03 of the time this cause occurred

Minimum Percentage for this type 90.00
Maximum Percentage for this type 99.99

Expert Robert changed current connection

The proposed connection threshold is 0.684010
This was determined based on the Mean 0.962121
    Less one Standard Deviation   0.178111
                Less SWAG factor   0.1
For this symptom when this cause occurred

Expert Robert decided to change threshold

Making a change to a connection, Refining Expert Knowledge

SYMPTOM TAILINGPEAKS

●Current Connection for Symptom TAILINGPEAKS
to Cause COLUMNDEGRADATION is a SOMETIMES

●Proposed Connection for Symptom TAILINGPEAKS
to Cause COLUMNDEGRADATION is a SOMETIMES

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 45.45 of the time this cause occurred

Minimum Percentage for this type 40.00
Maximum Percentage for this type 90.00

Confirming Connection Type

The proposed connection threshold is 0.000000
This was determined based on the Mean 0.283636
    Less one Standard Deviation   0.352055
                    Less SWAG factor   0.1
For this symptom when this cause occurred

Expert Robert accepted change to threshold
No change needed, connection remains as before

SYMPTOM UNRESOLVEDPEAKS

●Current Connection for Symptom UNRESOLVEDPEAKS
to Cause COLUMNDEGRADATION is a SOMETIMES

●Proposed Connection for Symptom UNRESOLVEDPEAKS
to Cause COLUMNDEGRADATION is a NEVER

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 100.00 of the time this cause occurred

Minimum Percentage for this type   0.00
Maximum Percentage for this type   0.00

Expert Robert changed current connection

The proposed connection threshold is 0.000000
This was determined based on the Mean 0.000000
    Less one Standard Deviation   0.000000
                    Less SWAG factor   0.1
For this symptom when this cause occurred

Expert Robert decided to change threshold
Making a change to a connection, Refining Expert Knowledge

SYMPTOM RISINGBASELINE

●Current Connection for Symptom RISINGBASELINE
to Cause COLUMNDEGRADATION is an ALWAYS

●Proposed Connection for Symptom RISINGBASELINE
to Cause COLUMNDEGRADATION is an INFREQUENTLY

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 96.97 of the time this cause occurred

Minimum Percentage for this type Non-zero
Maximum Percentage for this type 40.00

Expert Robert changed current connection

The proposed connection threshold is 0.000000
This was determined based on the Mean 0.030303
    Less one Standard Deviation   0.174078
                    Less SWAG factor   0.1
For this symptom when this cause occurred

Expert Robert decided to change threshold
Making a change to a connection, Refining Expert Knowledge

SYMPTOM HIGHBACKGROUND

●Current Connection for Symptom HIGHBACKGROUND
to Cause COLUMNDEGRADATION is a SOMETIMES

●Proposed Connection for Symptom HIGHBACKGROUND
to Cause COLUMNDEGRADATION is a NEVER

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 100.00 of the time this cause occurred

Minimum Percentage for this type 0.00
Maximum Percentage for this type 0.00

Expert Robert changed current connection

The proposed connection threshold is 0.000000
This was determined based on the Mean 0.000000
    Less one Standard Deviation   0.000000
                    Less SWAG factor   0.1
For this symptom when this cause occurred

Expert Robert decided to change threshold
Making a change to a connection, Refining Expert Knowledge

SYMPTOM LEADINGPEAKS

●Current Connection for Symptom LEADINGPEAKS
to Cause COLUMNDEGRADATION is a SOMETIMES

●Proposed Connection for Symptom LEADINGPEAKS
to Cause COLUMNDEGRADATION is a SOMETIMES

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 39.39 of the time this cause occurred

Minimum Percentage for this type 40.00
Maximum Percentage for this type 90.00

Confirming Connection Type
The proposed connection threshold is 0.000000

This was determined based on the Mean 0.468485
    Less one Standard Deviation   0.439724
                    Less SWAG factor   0.1
For this symptom when this cause occurred

Expert Robert accepted change to threshold
No change needed, connection remains as before

****
Done confirmation and refinement of expert knowledge
****

SYMPTOM SPIKEPRECISION

DELETION, symptom values always negative 2

SYMPTOM SURROGATEPRECISION

DELETION, symptom values always negative 2

SYMPTOM SENSITIVITYCHANGE

Currently there is no connection for Symptom SENSITIVITYCHANGE
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom SENSITIVITYCHANGE
to Cause COLUMNDEGRADATION is an ALWAYS

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 0.00 of the time this cause occurred

Minimum Percentage for this type 100.00
Maximum Percentage for this type 100.00

Expert Robert added this connection

The proposed connection threshold is 0.891825
This was determined based on the Mean
            Less one Standard Deviation Less 0.1
Mean 0.998788   Std Dev 0.006963 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

SYMPTOM BANDBROADENING

Currently there is no connection for Symptom BANDBROADENING
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom BANDBROADENING
to Cause COLUMNDEGRADATION is an INFREQUENTLY

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 78.79 of the time this cause occurred

Minimum Percentage for this type Non-zero
Maximum Percentage for this type 40.00

Expert Robert added this connection

The proposed connection threshold is 0.000000
This was determined based on the Mean
            Less one Standard Deviation Less 0.1
Mean 0.161212   Std Dev 0.330149 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

SYMPTOM CLIPPEDPEAKS

Currently there is no connection for Symptom CLIPPEDPEAKS
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom CLIPPEDPEAKS
to Cause COLUMNDEGRADATION is a NEVER

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 100.00 of the time this cause occurred

Minimum Percentage for this type 0.00
Maximum Percentage for this type 0.00

Expert Robert added this connection

The proposed connection threshold is 0.000000
This was determined based on the Mean
        Less one Standard Deviation Less 0.1
Mean 0.000000   Std Dev 0.000000 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

SYMPTOM IRREGULARBASELINE

Currently there is no connection for Symptom IRREGULARBASELINE
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom IRREGULARBASELINE
to Cause COLUMNDEGRADATION is a SOMETIMES

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 15.15 of the time this cause occurred

Minimum Percentage for this type 40.00
Maximum Percentage for this type 90.00

Expert Robert added this connection

The proposed connection threshold is 0.384375
This was determined based on the Mean
        Less one Standard Deviation Less 0.1
Mean 0.848485   Std Dev 0.364110 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

SYMPTOM GHOSTPEAKS

DELETION, symptom values always negative 2

SYMPTOM EXTRAPEAKS

Currently there is no connection for Symptom EXTRAPEAKS
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom EXTRAPEAKS
to Cause COLUMNDEGRADATION is a NEVER

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 100.00 of the time this cause occurred

Minimum Percentage for this type   0.00
Maximum Percentage for this type   0.00

Expert Robert added this connection

The proposed connection threshold is 0.000000
This was determined based on the Mean
          Less one Standard Deviation Less 0.1
Mean 0.000000   Std Dev 0.000000 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

SYMPTOM NOPEAKS

Currently there is no connection for Symptom NOPEAKS
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom NOPEAKS
to Cause COLUMNDEGRADATION is a NEVER

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 100.00 of the time this cause occurred

Minimum Percentage for this type   0.00
Maximum Percentage for this type   0.00

Expert Robert added this connection

The proposed connection threshold is 0.000000
This was determined based on the Mean
          Less one Standard Deviation Less 0.1
Mean 0.000000   Std Dev 0.000000 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

SYMPTOM REPLICATEPRECISION

Currently there is no connection for Symptom REPLICATEPRECISION
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom REPLICATEPRECISION
to Cause COLUMNDEGRADATION is an ALWAYS

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 0.00 of the time this cause occurred

Minimum Percentage for this type 100.00
Maximum Percentage for this type 100.00

Expert Robert added this connection

The proposed connection threshold is 0.883650
This was determined based on the Mean
          Less one Standard Deviation Less 0.1
Mean 0.997576   Std Dev 0.013926 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

SYMPTOM CANNOTZEROBASELINE

DELETION, symptom values always negative 2

SYMPTOM NEGATIVEDIPAFTERPEAK

Currently there is no connection for Symptom NEGATIVEDIPAFTERPEAK
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom NEGATIVEDIPAFTERPEAK
to Cause COLUMNDEGRADATION is a NEVER

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 100.00 of the time this cause occurred

Minimum Percentage for this type 0.00
Maximum Percentage for this type 0.00

Expert Robert added this connection

The proposed connection threshold is 0.000000
This was determined based on the Mean
       Less one Standard Deviation Less 0.1
Mean 0.000000   Std Dev 0.000000 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

- - - - - - - - - - - - - - - - - - - -
SYMPTOM IRREGULARSPIKES
- - - - - - - - - - - - - - - - - - - -
Currently there is no connection for Symptom IRREGULARSPIKES
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom IRREGULARSPIKES
to Cause COLUMNDEGRADATION is a NEVER

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 100.00 of the time this cause occurred

Minimum Percentage for this type 0.00
Maximum Percentage for this type 0.00

Expert Robert added this connection

The proposed connection threshold is 0.000000
This was determined based on the Mean
       Less one Standard Deviation Less 0.1
Mean 0.000000   Std Dev 0.000000 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

- - - - - - - - - - - - - - - - - - - -
SYMPTOM HIGHNOISE
- - - - - - - - - - - - - - - - - - - -
Currently there is no connection for Symptom HIGHNOISE
to Cause COLUMNDEGRADATION

●Proposed Connection for Symptom HIGHNOISE
to Cause COLUMNDEGRADATION is an INFREQUENTLY

Connection type determined by frequency of symptom occurrence
Symptom was <= 0 84.85 of the time this cause occurred

Minimum Percentage for this type Non-zero
Maximum Percentage for this type 40.00

Expert Robert added this connection

The proposed connection threshold is 0.000000
This was determined based on the Mean
Less one Standard Deviation Less 0.1
Mean 0.094545 Std Dev 0.267373 for this Cause

Expert Robert accepted proposed connection
Augmenting Expert Knowledge
Adding a connection that was not there

APPENDIX H

NETMEDIC: THE SOFTWARE

NetMedic software consists of three files: `NetMedic.c`; `NetMedic.h` and a `Makefile`. Written in ANSI C, it uses the following include files: `stdlib.h`; `stdio.h`; `time.h`; `math.h`; `string.h`; `ctype.h`; and `unistd.h`. The author compiled the software with the gcc compiler version 2.6.0 on a Sparc 10 running SunOS 4.1.4.

The main data structures utilized in NetMedic are in `NetMedic.h`. The structure `cause_node` represents all the information contained in one `bob-output` file generated by the CAEN network. NetMedic creates a linked list of these structures of type `cause_file_list_t`. This `cause_file_list_t` only contains two pointers, to the head and the tail of the list. The variable declared for this list of cause file information is `cause_file_list`.

The structure `cause_statistical_info` holds all the statistical information of one cause. The `cause_statistical_info` structure contains fields for how many times the cause occurred in the data set, how many times the network predicted this cause correctly, and the means and standard deviations of the cause values.

Another major component is the `symptom_statistics_array` of type `symptom_statistics_t`. The `symptom_statistics_array` component holds

all of the symptom statistics as this symptom relates to this cause. The `cause_statistical_info_array` in `NetMedic.c` uses the structure `cause_statistical_info` for each of its elements.

The last major data structure in `NetMedic.h` is the structure `connection_to_add_t`. It holds information about proposed connections for the network. A two dimensional array called `proposed_connections` found in `NetMedic.c` uses the `connection_to_add_t` structure. The software indexes the array first by the cause index and secondly by the symptom index.

The other major data structures are two arrays found in `NetMedic.c`. They provide arrays of cause and symptom names for easy lookup.

NetMedic's main program utilizes three main procedures: `get_input_file()`; `collect_cause_info()`; and `alter_architecture()`. All three are void procedures and return void. NetMedic's main calls `get_input_file()` in a for loop, once for each input file. NetMedic's main calls the other two procedures only once.

Procedure `get_input_file()` reads each input file and builds the main list structure called `cause_file_list`. It fills each list element, a `cause_node`, with the appropriate information from the input file. It also determines the number of causes found in the data set and the number of symptoms found in each file. The number of causes and the number of symptoms must remain constant during a given run.

The procedure `collect_cause_info()` builds the statistical information used by NetMedic to determine connection types. It produces a statistics file called `results.out`. The statistics file contains the various statistics developed by the procedure.

For each cause, `collect_cause_info()` traverses the `cause_file_list` list twice. The first sweep collects information relating to this cause to build up the means. The second sweep determines the standard deviation and correlation coefficient. These statistics break down into three main groups. The total group covers all information for cases when this cause was expected. The good group statistics cover all information for cases when this cause was expected and predicted correctly. The bad group statistics cover the complement of the set of cases covered in the good group. `Collect_cause_info()` determines the mean and standard of the cause and each symptom for these groups. In addition, for symptoms it determines the correlation coefficient and the number of times the symptom value was 0, was -1, was -2 or was positive.

`Alter_architecture()` is the workhorse of NetMedic. It repeats until either the user has modified all the causes or until the user opts to exit. It creates two files and requires one as input. The input file must be in the `common.net` format described in Appendix D. The `common.net` output file created will be in the same format. The other output file is a recording of the expert/user 's decisions during use of NetMedic.

`Alter_architecture()` calls a procedure `get_cause_to_fix()` to determine which cause the user wishes to modify or if the user wishes to exit. It then calls `set_proposed_connection_info()`, passing the cause index for each symptom known. The `set_proposed_connection_info()` procedure determines the connection types based upon the frequency of occurrence at non-zero or negative values. `Alter_architecture()` checks for all possible parallel paths already in the input file and then calls `make_common_net()` passing the cause index to be fixed.

`Make_common_net()` first reads the input file, transferring any comments to the output `common.net` file being created. It then processes each Symptom to Combination node entry one by one. It checks to see if the Combination node entry relates to the cause the user wishes to fix. If it does not, the procedure writes the entry to the output file. If it does, the procedure compares the connection to the proposed connection. If they agree, the procedure analyzes the weight to determine the importance to the network. If the weight is very small, the procedure prompts the user to delete the connection. The procedure also prompts the user to modify the threshold if he/she would like. If the two connections do not agree, the user is prompted to decide whether to change the connection, use the current connection in the input file or delete the connection altogether. The procedure queries the user about setting the threshold also.

Once `make_common_net()` finishes with the existing connections, it then prompts the user to decide about other proposed connections. NetMedic determines a connection for every symptom to every cause. Chapter 6 explains the reasons for making a proposal for each symptom. If the user decides to add a connection, the user must determine whether to accept a non-default threshold. The resultant choices become entries into the output file.

Once these other connections are made, the `make_common_net()` procedure copies over the Combination node to Cause node connections and exits.

APPENDIX I

DECEMBER 1994 KNOWLEDGE TABLE

The December 1994 knowledge table is the result of the Contaminant Analysis Expert Network (CAEN) team's effort in knowledge acquisition from May 1994 to December 1994. The creation of this knowledge table began at a meeting in Albuquerque in May 1994, and is based on Professor Martin Stillman's work with truth tables [Lahiri and Stillman]. The Albuquerque meeting contributors include: John Elling, LANL; John Robinson, Varian Equipment Corporation; John Feddema, SNL; Leon Klatt, ORNL; Martin Stillman and Hai Du, University of Western Ontario; Joel Matek and George Luger, University of New Mexico; and Alan Levis and Susan Hruska, Florida State University. The information is from experts only; NetMedic did not exist in December 1994.

The knowledge table utilizes the AUSIN entries described in this thesis. The table contains rows relating to the symptoms that may appear. The columns represent causes. The experts further grouped the causes into one of six main areas: Carrier Gas, Injection, Injector, Column, Fuel Gas, and Detector. These six areas relate to major component areas of the Gas Chromatography equipment.

The row and column intersections form cells. These cells may or may not have an entry, with a blank indicating no connection between symptom and cause. An "A" indicates that the symptom will always appear in a chromatogram if the given cause is present; a "U"

indicates a USUALLY relationship; an "S" indicates a SOMETIMES relationship; and an "I" indicates an INFREQUENTLY relationship; an "N" entry indicates a definite negative relationship between a given symptom and cause.

For example, for symptom *Retention Time Change* and cause *Late Elution* there is a "U" entry. This indicates the expert's opinion that if *Late Elution* occurs, the symptom *Retention Time Change* will usually appear in the chromatogram. For symptom *Retention Time Change* and cause *Column Bleed* there is an "N" entry. This indicates the expert's opinion that if *Column Bleed* occurs, the symptom *Retention Time Change* will definitely not appear in the chromatogram.

The shaded cause columns indicate the causes for which the CAEN team has actual data. The signal processing team has developed routines to extract from chromatograms those symptom values which are shaded.

CAA/DIM symptom/cause relationships

Adapted from DIM expert information and Varian, Diagnosis/Troubleshooting Guide

Dated December 1994

Goals: **Causes**

Symptoms	Injector 1. Leaking Syringe	2. Dirt in the Syringe	Column 3. Sample Decomposition	4. Late Elution	5. Column Bleed	6. Column Degradation	7. Contaminated Column	Detector 8. Contaminated Detector	9. Wrong Fuel Gas Flow (ECD)	10. Dust in Flame (FID)	11. Make Up Gas Loss (ECD)	Injection 12. Dirt in the Injector	13. Leaking Septum	Sample 14. Sample Too Concentrated	15. Sample Size Too Large	16. Sulphur Contamination	17. Phalate Contamination	Fuel Gas 18. Carrier Gas Low
1. Retention Time Change	A	U		U	N	A						I	S	S	N			A
2. Spike/Surrogate precision and reproducibility	A	U	S															
3. Sensitivity Change					N			A	A		A	S	A					U
4. Peak Tailing					N	S	S				S	S	N					
5. Unresolved Peaks						S	S				S	S	S	S				S
6. Band Broadening				A										A	S			S
7. Split Peaks - early																		
8. Split Peaks - late																		
9. Clipped Peaks														A				
10. Rounded Peaks														U				
11. Negative Dip After Peak (ECD only)							S	U			I							
12. Baseline drift - irregular							S	U			I							
13. Baseline drift - monotonic (isothermal mode only)																		
14. Baseline drift - rising (temp programming mode only)			U		A		U						S	A				
15. Baseline stepping																		
16. Baseline drift - sinusoidal																		
17. Cannot Zero Baseline					S		U	S										
18. High Background Noise								U				U						
19. Irregular spikes										A								
20. Regular Spiking			U									U						
21. Ghost Peaks/Extra Peaks	U	A		A			S					S	S	S				
22. No Peaks	I		U															
23. Replicate Precision and reproducibility	A		U															
24. Leading Peaks	A		S			S								U	U			I

APPENDIX J

MARCH 1995 KNOWLEDGE TABLE

The March 1995 knowledge table is the result of the Contaminant Analysis Expert Network (CAEN) team's further efforts in knowledge acquisition, a modification of the table contained in Appendix I. The CAEN team along with John Elling, John Robinson, Joel Matek, Randy Roberts and Sharbari Lahiri made the changes during a meeting in Tallahassee in March 1995.

The knowledge table utilizes the AUSIN entries described in this thesis. Again, shaded rows and columns in the table indicate symptoms and causes for which the CAEN team received actual data.

CAA/DIM symptom/cause relationships

Adapted from DIM expert information Varian, Diagnosis/Troubleshooting Guide

Dated March 1995

Goals:

Symptoms \ Causes	Injector		Column					Detector				Injection		Sample				Fuel Gas
	1. Leaking Syringe	2. Dirt in the Syringe	3. Sample decomposition	4. Late Elution	5. Column Bleed	6. Column Degradation	7. Contaminated Column	8. Contaminated Detector	9. Wrong Fuel Gas Flow (ECD)	10. Dust in Flame (FID)	11. Make Up Gas Loss (ECD)	12. Dirt in the Injector	13. Leaking Septum	14. Sample Too Concentrated	15. Sample Size Too Large	16. Sulphur Contamination	17. Phtalate Contamination	18. Carrier Gas Low
1. Retention Time Change		U	S	U	U	A						I	S	S	N			A
2. Spike/Surrogate precision and reproducibility																		
3. Sensitivity Change	A				T	A		A	A		A	S	A					U
4. Peak Tailing				A		S	S				S	S	N					
5. Unresolved Peaks						S					S		I	S				S
6. Band Broadening				A		S								A	S			S
7. Split Peaks - early																		
8. Split Peaks - late																		
9. Clipped Peaks														A				
10. Rounded Peaks														U				
11. Negative Dip After Peak (ECD only)								U			I							
12. Baseline drift - irregular								U			I							
13. Baseline drift - monotonic (isothermal mode only)							U											
14. Baseline drift - rising (temp. programming mode only)					A	S							S	A				
15. Baseline stepping					S	S							S					
16. Baseline drift - sinusoidal								S								U		
17. Cannot Zero Baseline			U				U	U				U						
18. High Background Noise										A								
19. Irregular spikes	U																	
20. Regular Spiking	A	A		A														
21. Ghost Peaks							S					S	S	S				
22. /Extra Peaks			U										I					
23. No Peaks	I		U												U			
24. Replicate Precision and reproducibility	A																	
25. Leading Peaks			S			S							S					I

BIBLIOGRAPHY

Y. L. Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed. San Mateo, CA: Morgan Kaufman Publishers, 1990, pp. 598-605.

J. W. Elling, L. N. Klatt, and W. P. Unruh, "Automated data interpretation in an automated environmental laboratory," in *Laboratory Robotics and Automation*, vol. 6, no. 2, pp. 73-78, 1994.

S. E. Fahlman and C. Lebiere, "The cascade correlation learning architecture," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed. San Mateo, CA: Morgan Kaufman Publishers, 1990, pp. 524-532.

W. Fang and R. C. Lacher. "Stack: A constructive network learning algorithm," in *Proceedings of the 6th Florida Artificial Intelligence Research Symposium*, pp. 223-227, 1993.

Gensym Corporation, *G2 Reference Manual Version 3.0*. Cambridge, MA: Gensym Corporation, 1993.

J. Giarratano and G. Riley, *Expert Systems: Principles and Programming Second Edition*. Boston, MA: PWS-KENT Publishing Company, 1994.

B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in Neural Information Processing Systems 5*, S. Hanson, J. Cowan, and C. Giles, Eds. San Mateo, CA: Morgan Kaufman Publishers, 1993, pp. 164-171.

S. J. Henkind and M. C. Harrison, "An analysis of four uncertainty calculi," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 5, IEEE 8822673, 1988, pp. 700-714.

S. I. Hruska, "Building expert networks that really fly: Computational issues," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 3, pp. 1487-1492, July 1994.

D. C. Kuncicky, Isomorphism of reasoning systems with applications to autonomous knowledge acquisition. Ph.D. dissertation, Florida State University, 1991.

D. C. Kuncicky, S. I. Hruska, and R.C. Lacher, "Hybrid systems: The equivalence of rule-based expert system and artificial neural network inference," in *International Journal of Expert Systems*, vol. 4, no. 3, pp. 281-297, 1992.

R. C. Lacher, "Expert networks: Paradigmatic conflict, technological rapprochement," in *Minds and Machines* , vol. 3, pp. 53-71, 1993.

R. C. Lacher, S. I. Hruska, and D. C. Kuncicky, "Back-propagation learning in expert networks," in *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 62-72, 1992.

S. Lahiri and M. J. Stillman, "Expert systems, diagnosing the cause of problem AAS data," in *Analytical Chemistry*, vol. 64, no. 4, pp. 283-291, 1992.

A. P. Levis, R. G. Timpany, W. E. Austad, J. W. Elling, J. J. Ferguson, D. A. Klotter, and S. I. Hruska. "Application of knowledge-based network processing to automated gas chromatography data interpretation," in *Applications and Science of Artificial Neural Networks*, vol. 2492, pp. 294-302, 1995.

R. J. Mockler and D. G. Dologite, *Knowledge-Based Systems: An Introduction to Expert Systems*. New York, NY: Macmillan Publishing Company, 1992.

M. J. Stillman, "Development of expert systems for analytical chemistry," in *Encyclopedia of Analytical Science*, 1993.

M. J. Stillman, G. Huang, S. Lahiri, and Q. Zhu, "Acexpert. Design and implementation of Acselect, Aaexpert and GC-Mexpert: Expert systems that aid in the analysis of environmental samples," in *Expert Systems World Congress Proceedings*, J. Liebowitz, Ed. New York, NY: Pergammon Press, 1991.

A. Zlatkis and C. F. Poole, "Electron capture theory and practice in chromatography," in *Journal of Chromatography Library*, vol. 20, 1981.

BIOGRAPHICAL SKETCH

Robert G. Timpany received the Bachelor of Science from Rensselaer Polytechnic Institute in May 1984 and entered the U.S. Army through the Reserve Officers Training Corps. Currently holding the rank of Captain, Robert's military education includes: Infantry Officer Basic Course; RANGER school; Light Leader Course; Jungle Warfare Training Course; Infantry Officer Advanced Course; Survival Evasion Resistance and Escape Course; Special Forces Officers Qualification Course; Arabic/Egyptian Functional Language Course and the Combined Arms and Services Staff School. Robert served as an Infantry Officer with the 7th Infantry Division (Light) including deployments to Panama, Honduras and a six month tour in the Sinai Desert, Egypt, as part of the Multi-National Forces and Observers. Robert commanded a Special Forces Military Free-Fall Operational Detachment Alpha from January 1990 until July 1993 in 5th Special Forces Group (Airborne). Robert spent 18 months during this time in the Middle East in Jordan, Kuwait and Saudi Arabia. Robert was the Senior U.S. Advisor to a 600 man Saudi Arabian Army Mechanized Infantry Battalion during Desert Shield and Storm. Robert speaks elementary Spanish and Arabic with a working knowledge of English. Robert is a member of Upsilon Pi Epsilon and Phi Kappa Phi. Most importantly, Bob is a very well-adjusted individual in today's society.